Titre: Title:	Fusion of deformable anatomical structures of human torso
Auteur: Author:	Qing Hu Liao
Date:	2004
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Liao, Q. H. (2004). Fusion of deformable anatomical structures of human torso [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7411/

UNIVERSITÉ DE MONTRÉAL

FUSION OF DEFORMABLE ANATOMICAL STRUCTURES OF HUMAN TORSO

QING HU LIAO
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
DÉCEMBRE 2004

Your file Votre référence
ISBN: 0-494-01358-3
Our file Notre référence
ISBN: 0-494-01358-3

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

FUSION OF DEFORMABLE ANATOMICAL STRUCTURES OF HUMAN TORSO

présenté par: LIAO Qing Hu

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. BILODEAU Guillaume-Alexandre, Ph.D., président

Mme. CHERIET Farida, Ph.D., membre et directrice de recherche

M. GUIBAULT François, Ph.D., membre et codirecteur de recherche

M. MARTEL Yves, M.Sc.A., membre

To my parents and wife

ACKNOWLEDGEMENTS

My deepest gratitude goes to Farida Cheriet, my supervisor. She helped me a lot in this thesis project. Without her wise guidance, I wouldn't be able to finish this work. I sincerely thank François Guibault — my co-advisor, he gave me many helpful suggestions and comments on the thesis. I also want to thank the members of my thesis committee, Bilodeau Guillaume-Alexandre and Martel Yves, for their patience in reading drafts of my thesis, and their valuable feedback.

Special thanks to Valérie Pazos. She provided me the experiments data from Sainte-Justine Hospital, and kindly clarified the tedious details. Thanks to Rui Zhang for many inciting conversations and good research ideas.

Finally, I would like to thank to my wife — Rui hua Jiang, for her support and sacrifices.

RÉSUMÉ

La scoliose idiopathique est une maladie qui survient à l'âge de l'adolescence et qui peut entraîner des déformations complexes de la colonne vertébrale, de la cage thoracique et de la forme externe du tronc, avec une prévalence plus forte chez les filles. Le port du corset est une forme de traitement non chirurgical qui vise la prévention de la progression de la scoliose. Un modèle biomécanique 3D du tronc, obtenu à partir d'une méthode de modélisation par éléments finis est utile pour simuler le mécanisme d'interaction physique entre le corset, le tronc humain, la cage thoracique et la colonne vertébrale, pour planifier la chirurgie et documenter la cause d'une scoliose.

Cependant, les éléments de base considérés dans les modèles développés sont limités à des ressorts et des poutres. Les tissus mous qui séparent la surface externe du tronc et les structures osseuses sous-jacentes ne sont pas modélisés. De même, la surface externe du tronc n'est pas considérée puisque la force simulant l'effet du corset est appliquée directement sur la cage thoracique. Ces approximations et simplifications ont limité la précision de la simulation et suggèrent une amélioration du modèle.

Afin de générer un maillage tétraédrique personnalisé au tronc humain incluant la colonne vertébrale, la cage thoracique, les tissus mous et la surface externe du tronc à partir de données acquises à l'aide de techniques non effractives plusieurs étapes sont nécessaires. Une des techniques non efferactive permettant l'acquisition de l'ensemble des structures du tronc est la technique de résonance magnétique. Cependant, cette technique est encore assez coûteuse et donc n'est pas disponible de façon routinière en clinique. La technique de tomographie axiale offre aussi la possibilité de visualiser en 3D les structures osseuses du tronc mais la forte dose de radiation infligée au patient ne permet pas l'acquisition complète de l'ensemble rachis-cage thoracique. Par conséquent, à l'hôpital Sainte Justine les structures osseuses du tronc sont obtenues par reconstruction radiographique ce qui limite de façon significative la dose de radiation. Et la surface externe du tronc est acquise à l'aide d'un système de vision active non effractif.

Cependant, la surface externe du tronc et les structures osseuses sous-jacentes sont acquises à différents instants, dans des salles différentes, utilisant des techniques d'ac-

quisition différentes. Une procédure de recalage doit être effectuée pour fusionner les modèles des structures internes et externes avant de pouvoir générer un maillage 3D du tronc humain.

L'objectif principal de ce projet est de recaler et fusionner les modèles interne et externes des structures du tronc en utilisant une transformation élastique basée sur une mise en correspondance d'un ensemble de repères.

Dans le cadre de ce projet, nous avons comparé plusieurs méthode de recalage basées sur un ensemble de repères ; en l'occurrence les transformations rigides, affines et les méthodes d'interpolation et d'approximation utilisant les Splines Plaques Minces. La distance moyenne bidirectionnelle point-surface (BPTSMD) a été utilisée comme métrique de comparaison afin de déterminer la transformation la plus optimale.

Dans le cadre de ce projet, nous avons effectué plusieurs comparaisons entre différentes méthodes de recalage sur un ensemble de données cliniques. Les résultats ont démontré que les transformations élastiques sont plus appropriées pour le recalage de la surface externe du tronc et des structures osseuses sous-jacentes.

Nous avons aussi développé une méthodologie pour la génération d'un maillage tétraédrique entre la surface externe du tronc et la surface des structures osseuses sous-jacentes. La méthode FastRBF a été utilisée pour optimiser les modèles surfaciques définissant les frontières du volume et le logiciel Netgen a été utilisé pour générer un maillage tétraédrique entre les deux modèles surfaciques.

ABSTRACT

Adolescent idiopathic scoliosis is a disease that generates complex morphological deformations of the interior spinal column, rib cage and external trunk shape. Finite Element simulation of spine deformity gives good promise to help in the development of surgery plans; brace design and scoliosis cause investigation. However, previous Finite Element simulations were based on a coarse mesh model of the human skeletal trunk, which used beams and springs as basic elements and did not consider soft tissue and trunk skin. Brace forces were applied directly on the rib cage. This coarse model limits the simulation precision and a new finer mesh model is needed. In order to construct a personalized tetrahedron mesh model of human trunk including spinal column, rib cage, soft tissue and external trunk skin surface from clinical datasets via less-invasive 3D acquisition techniques, many steps of work are needed.

Among currently available techniques for acquiring 3D structures of the internal human skeleton, MRI can provide precise data but it is too expensive; and the affordable and efficient CT technique is hazardous to human body for providing large-scale 3D information such as human torso. In this project, the 3D internal bone structure model is reconstructed and provided by Sainte-Justine Hospital using the complex but cheap and less-invasive multiple-view X-ray technique. The 3D external trunk model is acquired using Inspeck's 3D optical camera system with high precision. Due to current technology limitations, the external and internal models are acquired at different times, in different sites and with different optimal postures. So we need to perform an elastic registration procedure before fusing them together.

The specific objective of this project is to align and fuse the external and internal human trunk surface models. We have investigated and compared several landmark-based registration methods including rigid, affine, Thin-Plate Spline (TPS) interpolation and approximation. We use the Bidirectional Point-to-Surface Mean Distance (BPTSMD) metric to estimate the deformation errors from the difference of postures. Lots of comparisons have been done and the results show that the local elastic deformation (TPS interpolation and approximation) methods are much better than the global transformation (rigid, affine) and suitable for registering the flexible external and internal models

of the human trunk. And we also have used Brent Minimization algorithm to find the optimal trade off parameter for the TPS approximation to account for the landmark localization errors.

In order to produce valid Boundary representation (B-Rep) of the multiple volume domain of human trunk, which is needed by most 3D tetrahedral mesh generation algorithms, we use the FastRBF (Fast Radial Basic Function) to handle the external trunk surface model (including hole-filling, smooth, simplification). Due to complex anatomical structures of internal bones, a lot of tedious works was done manually to produce a watertight volume representation of the bone structure. At last we use the multiple volume domain mesh generation algorithm based on NetGen to produce a coarse 3D tetrahedron mesh of the human trunk.

In this project, we also have done an interesting experiment to predict the external trunk shape change after scoliosis surgery. This prediction algorithm uses the TPS interpolation, and with the anatomical feature points on each vertebra before and after surgery as the corresponding landmarks. It is very simple, fast and can be used to help surgery plan and simulation.

CONDENSÉ

Introduction

La scoliose idiopathique est une maladie qui survient à l'âge de l'adolescence et qui peut entraîner des déformations complexes de la colonne vertébrale, de la cage thoracique et de la forme externe du tronc, avec une prévalence plus forte chez les filles. Le port du corset est une forme de traitement non chirurgical qui vise la prévention de la progression de la scoliose. Pour étudier l'effet d'un traitement par corset, un modèle biomécanique 3D du tronc, obtenu à partir d'une méthode de modélisation par éléments finis (Aubin99), est requis pour simuler le mécanisme d'interaction physique entre le corset, le tronc humain, la cage thoracique et la colonne vertébrale. Cette simulation par éléments finis est aussi utile pour la planification de la chirurgie (Aubin97b) et la documentation de la cause de la scoliose (Garceau02). Les activités de recherche antérieures impliquant la simulation par éléments finis sont basées sur le modèle d'éléments finis développé par Aubin et coll. (Aubin95). Cependant, ce modèle est approximatif vu les techniques et les performances de calcul disponibles durant cette période. Les éléments de base considérés dans ce modèle sont des ressorts et des poutres. Les tissus mous qui séparent la surface externe du tronc et les structures osseuses sous-jacentes ne sont pas modélisés. De même, la surface externe du tronc n'est pas considérée puisque la force stimulant l'effet du corset est appliquée directement sur la cage thoracique. Ces approximations et simplifications ont limité la précision de la simulation et suggèrent une amélioration du modèle.

Pour améliorer la précision de la simulation, nous avons besoin de développer un modèle biomécanique 3D personnalisé du tronc humain, basé sur un maillage tétraédrique, intégrant les os, les ligaments et les tissus mous obtenus à partir de données cliniques acquises à l'aide de systèmes non effractifs. Cependant, la surface externe du tronc et les structures osseuses sous-jacentes sont acquises à différents instants, dans des salles différentes, utilisant des techniques d'acquisition différentes. La surface externe du tronc a été numérisée à l'aide d'une caméra optique 3D de la compagnie Inspeck. Alors que le modèle 3D des structures osseuses a été obtenu par reconstruction 3D à partir d'images à rayons-X (Delorme99). Une procédure de recalage doit être effectuée pour fusionner

les modèles des structures internes et externes avant de pouvoir générer un maillage 3D du tronc humain. L'objectif principal de ce projet est de recaler et fusionner les modèles interne et externe des structures du tronc en utilisant une transformation élastique basée sur un ensemble de repères. Un ensemble de 20 marqueurs radio opaques sont collés sur la peau du patient et leur position 3D peut être identifiée sur les deux modèles ce qui permet d'établir la relation de mise en correspondance requise pour la procédure de recalage. Dans le cadre de ce projet, nous avons comparé plusieurs méthodes de recalage basées sur un ensemble de repères ; en l'occurrence les transformations rigides, affines et les méthodes d'interpolation et d'approximation utilisant les Splines Plaques Minces (Thin Plate Splines) (Rohr01). Nous avons développé un algorithme pour déterminer la valeur optimale du paramètre qui régit le schéma d'approximation dans le contexte de modélisation de la surface externe du tronc humain.

Nous avons aussi développé une méthodologie pour générer un maillage 3D tétraédrique entre la surface externe du tronc et la surface des structures osseuses sous-jacentes. Le principal problème est que la surface externe du tronc obtenue à partir des caméras Inspeck et la surface des structures osseuses obtenue à partir d'une reconstruction radiographique 3D ne constituent pas une représentation frontalière valide (B-Rep) du volume qu'elles délimitent. Les deux surfaces contiennent des trous, des arêtes non 'manifold' et des facettes triangulaires qui s'intersèctent. Nous avons proposé un ensemble d'algorithmes pour générer une représentation volumique simple et lisse. Nous avons aussi implémenté un algorithme de génération d'un maillage tétraédrique 3D pour le tronc humain basé sur le logiciel Netgen Cependant, le maillage 3D obtenu est juste une approximation initiale étant donné que nous avons appliqué plusieurs transformations de simplification sans prendre en considération la structure anatomique du tronc.

Enfin nous avons évalué le modèle des Splines Plaques Minces proposé en stimulant le résultat d'une chirurgie sur la surface externe du tronc connaissant l'effet de la chirurgie sur les structures osseuses du tronc.

État de l'art

Techniques d'acquisition de données 3D

Il existe plusieurs méthodes d'acquisition de la morphologie 3D des structures osseuses du tronc. La technique de résonance magnétique est une technique non effractive qui combinée avec des techniques de rendu permet une visualisation 3D précise des structures osseuses. Cependant, cette technique reste encore assez coûteuse pour une utilisation routinière en clinique. La technique de tomographie axiale permet aussi une visualisation 3D précise des structures osseuses mais la dose de radiation infligée au patient est trop importante. Pour des raisons d'éthique, on ne peut pas exiger une tomographie axiale du complexe rachis-cage thoracique et bassin. C'est pour ces raisons que les cliniciens se fient généralement aux images à rayons-X 2D pour établir un diagnostic ou prescrire un traitement. Notre groupe de recherche a développé une technique de reconstruction 3D qui permet de visualiser en 3D les structures osseuses à partir de deux ou trois radiographies (Delorme99). Cette méthode fournit un modèle surfacique des structures à partir de l'information personnalisée extraite des radiographies et un modèle générique d'un spécimen cadavérique obtenu par tomographie axiale. La précision de cette technique a été évaluée à $3.5 \pm 4.1mm$ ce qui est acceptable pour différentes applications. Cette technique est utilisée de façon régulière à l'hôpital Sainte Justine pour évaluer les déformations scoliotiques en 3D.

Deux facteurs importants doivent être pris en considération lors de l'acquisition de la surface externe du tronc. Le premier est qu'une précision adéquate du modèle est requise pour l'évaluation clinique de la déformation. Le deuxième est que la procédure d'acquisition doit être complétée dans un délai raisonnable afin d'éviter les changements de forme dues au mouvement du patient.

Il y a quelques années, une équipe de chercheurs à Calgary utilisait un système au laser (Poncet99) pour l'acquisition de la surface externe du tronc. L'erreur de mesure a été évaluée à 1.5 mm sur des objets rigides. Cependant, la procédure d'acquisition du tronc humain nécessite 10 secondes ce qui peut entraîner une détérioration de la précision. Notre groupe de recherche à Montréal utilise un système de caméras 3D de la compagnie Inspeck (www.inspeck.com) basé sur la projection de franges de moiré déphasées

sur la surface externe du tronc. Les caméras 3D de Inspeck permettent non seulement l'acquisition de la géométrie 3D d'un objet mais aussi sa couleur (information de texture) qui est combinée au modèle 3D obtenu. L'acquisition du tronc humain nécessite environ 3 secondes avec une précision de 1.1 mm et une résolution de 65 000 points ou 130 000 triangles ce qui fournit un maillage surfacique lisse et uniforme.

Techniques de recalage basées sur la mise en correspondance de repères

Étant donné que la surface externe du tronc et la surface des structures osseuses sous-jacentes ont été acquises à des instants différents, dans des salles différentes par deux modalités d'imagerie différentes une procédure de recalage s'impose avant la fusion des deux modèles surfaciques. Cette procédure de recalage peut être basée par deux type de tranformations : globale ou locale. Une transformation globale affine est une combinaison d'une mise à l'échelle isotropique, une translation et une rotation. Elle s'exprime comme suit : $p = sRq + t$ avec s un scalaire, R est la matrice de rotation, et t est le vecteur de translation. Une métrique basée sur la méthode des moindres carrés est utilisée pour trouver la solution optimale ; cette méthode est discutée en détail dans (Horn87). Dans le cas d'une transformation rigide le facteur d'échelle s est à un.

La technique d'interpolation utilisant les Splines Plaques Minces (Bookstein89, Evans91) peut prendre en considération des déformations locales élastiques en utilisant une métrique basée sur l'énergie d'inflexion. Celle-ci peut être exprimée à l'aide de la fonction suivante :

$$J_m^d(u) = \sum_{t_1+\ldots+t_d=m} \frac{m!}{t_1! \ldots t_d!} \times \int_{\mathbb{R}^d} \left(\frac{\partial^m u}{\partial x_1^{t_1} \ldots \partial x_d^{t_d}}\right)^2 d\boldsymbol{x} \qquad (1)$$

avec u est la fonction de transformation, m est l'ordre de dérivation, qui dépend de d la dimension du domaine. En général, deux noyaux de transformation peuvent être utilisés u : noyau r correspond au cas ou $m = 2, d = 3$; alors que le noyau r^3 correspond au cas $m = 3, d = 3$. Dans ce dernier cas on considère la troisième dérivée de la fonction d'énergie d'inflexion ce qui introduit un facteur de lissage global.

Avec la fonction d'énergie d'inflexion décrite précédemment, les repères sont mis en correspondance par superposition exacte. Une extension de cette approche serait l'intro-

duction d'un terme définissant une somme par moindres carrés afin de prendre en compte les erreurs de localisation de l'ensemble des repères considérés (Rohr01).

$$\boldsymbol{J}_\lambda(\boldsymbol{u}) = \frac{1}{n}\sum_{i=1}^{n}\frac{|\boldsymbol{q}_i - \boldsymbol{u}(\boldsymbol{p}_i)|^2}{\sigma_i^2} + \lambda \boldsymbol{J}_m^d(\boldsymbol{u}). \tag{2}$$

avec p_i, q_i l'ensemble des repères source et cible, et $\sigma_i^2 = \sigma_{i,p}^2 + \sigma_{i,q}^2$ est la somme des variances des erreurs de localisation des repères. Si le paramètre λ est petit, on obtient une transformation avec une petite déformation locale. Et si λ est grand, on obtient une transformation avec une grande déformation locale. Une technique d'interpolation correspond à une valeur intermédiaire de λ.

La technique des Splines Plaques Minces a été déjà utilisée avec succès pour le recalage de structures anatomiques obtenues par différentes modalités d'imagerie. Dans le cadre de ce projet, nous avons utilisé cette technique pour recaler la surface externe du tronc et la surface des structures osseuses sous-jacentes afin de générer un maillage 3D entre les deux surfaces.

Remplissage des trous et optimisation du maillage

Étant donné que les algorithmes de génération de maillage 3D nécessitent une représentation frontalière valide, un prétraitement de la surface externe du tronc et de la surface des structures osseuses sous-jacentes était nécessaire. Ce prétraitement consiste à remplir les trous et à éliminer les intersections et les arêtes non 'manifold' (une aréte qui est partagée par plus de deux triangles).

Étant donné la complexité de la géométrie et de la topologie des deux modèles surfaciques du tronc, un algorithme naïf de triangulation, qui nécessite une topologie en disque et des arêtes aux frontières planes, ne peut pas garantir un remplissage efficace des trous. Un algorithme basé sur une représentation surfacique implicite modélisée par la fonction FastRBF (Radial Basic Functions) (Carr01) a été utilisé. Les fonctions RBF sont populaires dans l'interpolation de données vu que le système d'équations linéaires associé est facile à inverser sous certaines conditions. La surface originale peut être représentée par une fonction de distance implicite suite à la résolution d'un système d'équations linéaires. La grande dimension du système nécessite l'utilisation de la mé-

thode Fast Multipode (FMM) qui combinée avec un algorithme itératif GMRES peut être utilisée pour résoudre ce système en un temps $O(k * N)$ avec $k << N$ (Beaston99). Ensuite la surface peut être re-polygonalisée à partir de cette fonction de distance implicite à l'aide de l'algorithme régularisé du 'Marching Tetrahedron' combiné avec une stratégie de suivi de la tangente. Dans cette procédure de re-maillage plusieurs paramètres peuvent être ajustés pour contrôler la précision et la résolution. Ainsi, nous pouvons remplir les trous et lisser la surface originale en même temps. Le logiciel commercial FastRBF a été fourni par FarField Technology Inc.

Algorithmes de génération de maillages 3D

Un maillage est dit structuré si tous les noeuds internes du maillage possèdent le même nombre d'éléments adjacents. Les éléments de base sont des quadrilatères en 2D et des éléments hexaédriques en 3D. Alors que dans un maillage non structuré un ensemble d'éléments peuvent converger vers un seul noeud. Les éléments triangulaires et tétraédriques sont les éléments les plus utilisés dans un maillage non structuré. Durant la dernière décennie, plusieurs techniques de génération de maillages 3D non structurés ont été proposées. La plupart d'entre elles sont basées sur le critère de Delaunay (Delaunay34). Dans des applications basées sur les éléments finis des conditions aux frontières doivent être imposées ce qui implique le principe de "Boundary Constrained Triangulation". Plusieurs méthodes ont été proposées pour satisfaire ce critère (Weatherill94, George91, Joe91). Un autre algorithme populaire de génération de maillage est celui basé sur la méthode Advancing Front (Lohner98, Lohner96).

Actuellement, plusieurs logiciels de génération de maillage non structurés sont disponibles. Netgen est un logiciel gratuit distribué sous une licence LGPL (Schoberl04) ce qui nous permet de le modifier et de l'adapter à nos besoins.

Matériel et méthode

Acquisition des données cliniques

Les données utilisées proviennent de l'Hôpital Sainte Justine et ont été acquises pour le suivi des patients scoliotiques. Les postures adoptées par les patients ont été déterminées selon le système d'imagerie utilisé afin de garantir la visibilité des structures d'intérêt

sur les images acquises. Par conséquent, les postures adoptées pour l'acquisition de la surface externe du tronc et des structures osseuses sous-jacentes sont différentes. Soit A la posture adoptée pour l'acquisition des radiographies et B la posture adoptée pour l'acquisition de la surface externe du tronc.

Pour fixer la valeur des paramètres de la technique de recalage proposée, nous avons acquis quatre modèles de surface externe pour chaque patient, deux modèles à partir de la posture A et les deux autres à partir de la posture B. Cependant, un seul modèle de la surface des structures osseuses acquis à partir de la posture A était disponible pour chaque patient. Une vingtaine de marqueurs radio opaques ont été collés sur la surface externe du tronc avant l'acquisition des données. La position 3D de l'ensemble des marqueurs peut être identifiée sur les deux modèles surfaciques. Ces marqueurs ont été utilisés dans la procédure de recalage et de fusion des deux modèles.

Technique de recalage

Les deux modèles surfaciques obtenus étant représentés dans deux systèmes de coordonnées différents, ces derniers doivent être recalés avant d'être fusionnés. Comme le tronc humain est de nature déformable une transformation élastique devrait être plus appropriée. Cependant, la précision de la fusion est difficile à évaluer puisque les deux modèles représentent des structures différentes. Nous avons alors évalué l'erreur de superposition de deux modèles de surface externe du tronc obtenu à partir de deux postures différentes.

Un modèle surfacique externe obtenu à partir de la posture A est considéré comme le modèle de référence (le modèle cible) et le modèle surfacique obtenu à partir de la posture B (modèle source) doit être transformé afin d'être superposé au modèle cible. Plusieurs types de transformation ont été utilisées et le modèle surfacique transformé a été comparé au modèle cible en utilisant la même métrique.

La distance moyenne bidirectionnelle point-surface (BPTSMD) a été utilisée comme métrique de comparaison afin de déterminer la transformation la plus optimale. La métrique BPTSMD est facile à calculer et nous permet de visualiser la distribution des erreurs sur les différentes régions du tronc (Figure 2.6). Étant donné la grande taille du modèle surfacique du tronc (60,000 points, 130,000 triangles), un algorithme basé sur

les 'OCTREE' a été utilisé pour accélérer le calcul.

Pour la méthode d'approximation utilisant les Splines Plaque Minces la valeur optimale pour le paramètre λ doit être déterminée. Les expériences effectuées suggèrent que la valeur optimale du paramètre λ se situe dans l'intervalle (-5, 1). Nous avons supposé que les erreurs de localisation de la position 3D des repères est dans l'ordre de la résolution du système d'acquisition.

Prétraitement des modèles surfaciques

La méthode FastRBF a été utilisée pour remplir les trous et optimiser le maillage surfacique externe du tronc. Cependant, la méthode FastRBF nécessite un maillage surfacique consistent en entrée. Les arêtes connectées à plus de deux triangles (non manifold) doivent être d'abord enlevées. Étant donné la complexité de la topologie cette procédure introduit d'autres trous dans la surface. Un algorithme agressif qui élimine les arêtes non manifold et les triangles associés a été utilisé même si cela produit des maillages surfaciques déconnectés avec des trous plus importants. Les plus grands maillages sont soumis comme entrée à la méthode FastRBF pour générer un maillage surfacique valide.

Le modèle surfacique des structures osseuses obtenu à partir des radiographies a été généré pour des fins de visualisation 3D. Par conséquent, le maillage obtenu ne représente qu'une approximation initiale qui est loin de vérifier les propriétés de validité. Un travail manuel laborieux a été effectué afin de transformer le maillage initial en une représentation frontalière 'Watertight' valide.

Résultats et discussion

Un échantillon de 13 patients scoliotiques a été utilisé dans notre étude. Pour chaque patient, quatre modèles surfaciques du tronc (deux en posture A et deux en posture B) et un modèle surfacique des structures osseuses (en posture A) ont été acquis. Pour diverses raisons 11 patients seulement possèdent quatre modèles surfaciques du tronc exploitables, un patient possède uniquement trois et pour un patient seulement nous avons les modèles surfaciques du tronc et des structures osseuses sous-jacentes avant et après la chirurgie.

Table 1 Résultats des différentes méthodes de recalage.

Patient ID	**Samples**	$rigid$	$affine$	tps_{in}	$tps_{in}(r^3)$	tps_{ap}	λ
F0003	8	3.423	3.131	2.864	3.594	2.829	-0.31 $\pm$ 0.951
F0006	8	3.666	3.670	3.016	4.442	3.010	-0.07 $\pm$ 0.234
G0007	8	3.153	3.038	2.651	3.064	2.635	0.10 $\pm$ 0.390
F0008	8	4.048	3.935	3.718	4.289	3.699	-0.48 $\pm$ 0.991
F0014	4	6.213	4.283	3.626	3.281	3.574	0.12 $\pm$ 0.393
G0015	8	5.815	4.859	4.097	5.402	4.053	-0.80 $\pm$ 0.548
F0016	8	2.748	3.154	3.041	3.799	3.004	-1.66 $\pm$ 1.601
F0017	8	4.638	4.264	2.808	3.921	2.804	0.06 $\pm$ 0.123
F0025	8	8.305	7.003	3.733	3.649	3.572	0.57 $\pm$ 0.153
F0026	8	4.071	3.916	3.183	3.652	3.174	0.04 $\pm$ 0.198
F0027	8	3.755	4.316	3.505	4.037	3.439	0.44 $\pm$ 0.075
F0029	8	4.258	3.584	3.118	3.081	3.071	0.14 $\pm$ 0.196
Total	92	4.434	4.088	3.265	3.876	3.224	-0.17 $\pm$ 0.863
Ratio: $1/tps_{ap}$		135.5%	126.8%	101.3%	120.2%	100%	

Le recalage de deux modèles surfaciques du tronc acquis à partir des deux postures A et B a été effectué en utilisant les transformations rigide, affine, et interpolation et approximation par Splines Plaques Minces. La métrique BPTSMD a été calculée afin de comparer le modèle transformé de la posture B au modèle acquis selon la posture A. Un total de 92 comparaisons a été effectué. Pour la méthode d'approximation par Splines Plaques Minces l'algorithme de minimisation de Brent a été utilisé pour obtenir la valeur optimale. Les résultats de ces expériences sont présentés dans le Table 1.

Le Table 1 montre que les résultats obtenus par les méthodes d'interpolation et d'approximation par Splines Plaques Minces utilisant un noyau de r sont toujours meilleurs par rapport à ceux obtenus à l'aide de méthodes de transformation globale rigide et affine. Les résultats obtenus à l'aide d'une méthode d'interpolation utilisant un noyau r^3 ne sont pas intéressants car le facteur de lissage introduit est trop important. Nous pouvons constater que la méthode d'approximation fournit toujours les meilleurs résultats de recalage. Cependant, la valeur optimale du paramètre λ est incertaine (-0.17 ± 0.863).

La méthodologie développée pour la génération d'un maillage tétraédrique du tronc a été évaluée sur un seul patient. Cependant, le maillage 3D obtenu est juste une approximation initiale et plusieurs améliorations doivent être effectuées avant de l'utiliser dans des applications de simulation de traitement basées sur des modèles par éléments finis.

Conclusion et perspectives

Dans le cadre de ce projet, nous avons effectué plusieurs comparaisons entre différentes méthodes de recalage sur un ensemble de données cliniques. Les résultats ont démontré que les transformations élastiques sont plus appropriées pour le recalage de la surface externe du tronc et des structures osseuses sous-jacentes. La distance moyenne bidirectionnelle point-surface a été utilisée comme métrique pour comparer les résultats de recalage des différentes méthodes.

Nous avons aussi développé une méthodologie pour la génération d'un maillage tétraédrique entre la surface externe du tronc et la surface des structures osseuses sous-jacentes. La méthode FastRBF a été utilisée pour optimiser les modèles surfaciques définissant les frontières du volume et le logiciel Netgen a été utilisé pour générer un maillage tétraédrique entre les deux modèles surfaciques.

Dans le cadre de travaux futurs il est souhaitable d'étudier l'impact de la configuration des marqueurs sur les résultats du recalage. De plus, une estimation des erreurs de localisation des marqueurs serait utile car actuellement une erreur isotropique équivalente à la résolution des caméras 3D a été considérée. Une métrique standard (distance de Hausdorf) devrait être utilisée pour évaluer les différentes méthodes de recalage. Une validation de l'approche proposée doit être effectuée sur un échantillon plus significatif de patients.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

$3D$: 3 Dimension

$BPTSMD$: Bidirectional Point-to-Surface Mean Distance

FE : Finite Element

RBF : Radial Basic Function

SR : Simplification Ratio

tps : Thin-Plate Spline

tps_{ap} : Thin-Plate Spline Approximation (kernel: r)

tps_{in} : Thin-Plate Spline Interpolation (kernel: r)

$tps_{in}(r^3)$: Thin-Plate Spline Interpolation (kernel: r^3)

LIST OF APPENDIXES

INTRODUCTION

Scoliosis is a disease that generates complex morphological deformations of the internal spinal column, rib cage and external trunk shape. In most cases, the cause of scoliosis is unknown and such cases are called idiopathic scoliosis. This is particularly so among the type of scoliosis seen in adolescent girls. The goal of scoliotic treatment is to stop the progression of the spinal column curve and prevent progression of the deformity. Treatment may include observation, repeated examinations, bracing, surgery and other approaches. Treatment choice in adolescent idiopathic scoliosis (AIS) is determined by a complex equation that includes the patient's physiologic maturity, curve magnitude and location and potential for progression. According to past research, two or three persons in every 1000 will need active treatment for a progressive condition (Weinstein84). In one out of every 1000 cases, surgery may be necessary (NSF02).

Due to the prevalence of this disease, many efforts have been spent on the research of spinal deformity from different points of view. In recent years, the Finite Element simulation method has been introduced into this field. It could be used to help the development of brace design by simulating the spinal deformity and studying the physical interaction mechanism between brace, external trunk, soft tissue, rib-cage and spine column (Aubin99). Also, Finite Element simulation is helpful for surgery plan (Aubin97b) and scoliosis cause investigation (Garceau02).

However, the previous research works on the Finite Element simulation of spinal deformity were based on the trunk skeletal mesh model created by Aubin et al (Aubin95) (Figure 1). This model was coarse due to the techniques and hardware available at that time. This model just represented the skeletal trunk with beam and spring as the basic elements. Therefore it didn't consider the soft tissue and external trunk skin. Many important factors are omitted in the simulation procedure based on this coarse mesh model. For example, in the simulation of brace treatment, the pressure forces of the brace are performed directly on the internal rib cage instead of external trunk skin surface. And the internal soft tissue that have strong relationship with the spinal deformity isn't considered into the simulation procedure. These shortcomings limited the simulation precision. With the development of modern 3D model acquisition techniques and

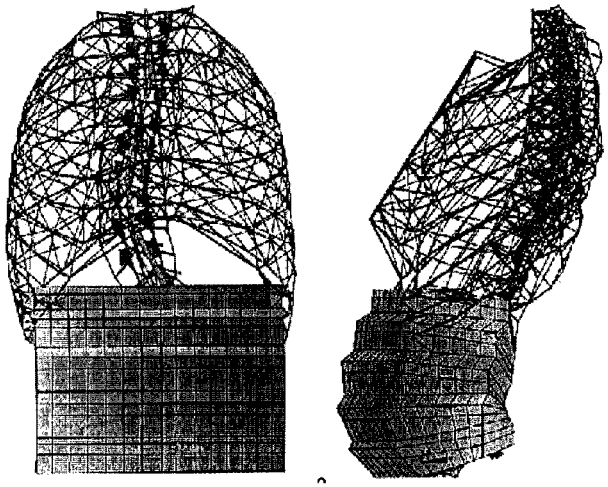


Figure 1 The previous skeletal trunk model created by Aubin et al for Finite Element simulation of spinal deformity (Aubin95). The vertebrae and rib cage are represented by beam elements; muscle and ligament are modeled using spring elements; The abdomen region are filled with hexhedrons.

the increasing computation ability of the commodity computers, creating a new more precise 3D trunk mesh model is possible and necessary to improve the Finite Element simulation precision of spinal deformity treatment.

The final objective of our research is fast and semi-automatically constructing a more accurate personalized 3D tetrahedron mesh of the human torso including the internal spinal column, rib cage, ligament, soft tissue and the external trunk skin from clinical data via less-invasive 3D acquisition techniques. The 3D external trunk skin surface model is digitized using Inspeck's 3D cameras. The 3D internal bone surface model is reconstructed from multiple X-ray images (Dansereau93). All these clinical models data are provided by Saint Justine Hospital, where these data are routinely collected for the purpose of diagnosis, brace design and surgery plan. This task is an extremely complex and hard. And it needs many researchers to collaborate and contribute for each steps.

In this project, we focus on registering and fusing the flexible external and internal human trunk models by local elastic deformation methods. At the same time, we would like to do some method investigation and development works to find the suitable ways towards our final objective. Three main steps of our work are listed as the following.

Registration: The external trunk skin and internal bone surface models are acquired using different sensors at different times, sites and even with different postures. Since the human body is a flexible objective, an elastic registration procedure must be performed before fusing these models into the same coordinate system space. In this project, we have investigated and compared several landmark-based registration methods. Specifically the Thin-Plate Spline Interpolation (tps_{in}) and Approximation (tps_{ap}) method (Rohr01) have showed much better results than the rigid method. We have designed an efficient algorithm to find the optimal trade off parameter of tps_{ap} in the context of the human trunk.

Surface Mesh Preprocessing: A valid volumetric representation of the human trunk and bone is necessary for 3D mesh generation. However, the 3D external trunk skin and internal bone surface models acquired from Inspeck's 3D cameras and Multiple X-ray images aren't valid boundary representations (B-Rep) of the corresponding volumes. There are holes, non-manifold edges and self-intersecting triangles in these surface models. There are too many points and cells, and the surface is not smooth. We propose a series of algorithms and techniques to produce valid, precise, simplified, smooth and watertight surface meshes for the external trunk skin and internal bone models.

3D Mesh Generation: At last, a multiple domain mesh generation algorithm based on the open source software NetGen is used to generate the 3D tetrahedron mesh for the whole human trunk.

The first part is the most important for our project and we will do many detailed evaluation works to verify the registration results. The remaining steps are just some initial experiments for the method development towards the final objective. See Figure 2 for the whole procedure that is needed to generate the simplified personalized 3D tetrahedron mesh of the human trunk from the clinical models data.

In the scoliotic surgery, the instruments are often used to correct the position of vertebrae to get rid of the spinal column curve and prevent progression of deformity. A simple and fast algorithm based on the Thin-Plate Spline interpolation is developed to predict the external trunk shape deformation after the surgery. Here we use the anatomical features points on vertebrae as landmarks.

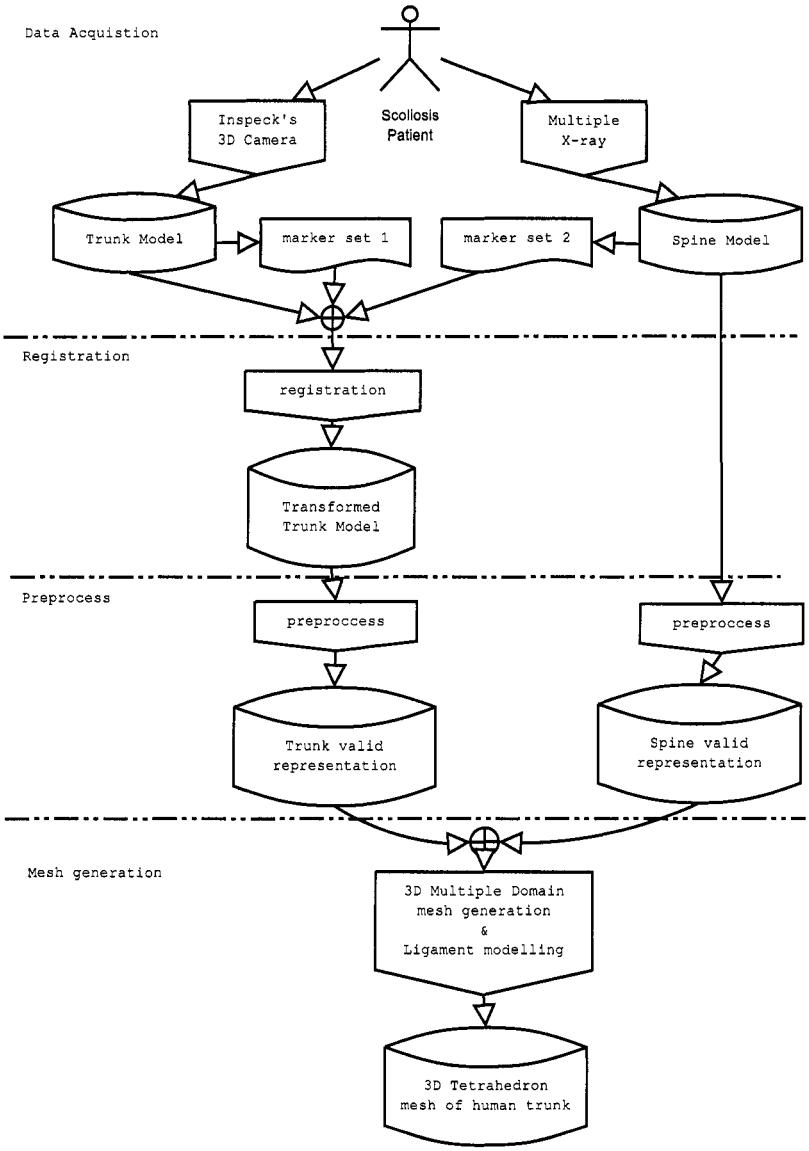


Figure 2 The whole procedure of the 3D mesh generation for the human trunk from clinical external trunk skin and internal bone surface models.

This thesis is organized as follows:

1. Chapter 1 — State of Art. First we review the state of art of the 3D model acquisition techniques, rigid and elastic registrations, hole filling techniques and 3D mesh generation algorithms. Then we describe our research objective in the last section.

2. Chapter 2 — Material and Method. In section 2.1, we talk about the clinical external and internal models acquisition procedure and protocol. Then we design and implement an evaluation metric to compare the registration results and compute the optimal trade off parameter for elastic approximation scheme in section 2.2. And we introduce the surface mesh preprocessing procedure and multiple domain 3D tetrahedral mesh generation in section 2.3 and 2.4 respectively. In section 2.5 we talk the external trunk shape prediction algorithm.

3. Chapter 3 — Result and Discussion. In this chapter, we will show our experimental results and make some discussion.

4. Chapter 4 — Conclusions and future works.

CHAPTER 1

THE STATE OF ART

In this chapter, first we review the state of art of the algorithms and techniques of 3D model acquisition, landmark-based registration, hole filling, 3D mesh generation. Then we propose our research objective in the last section.

1.1 3D Acquisition Techniques

To generate the 3D tetrahedral mesh model of the human trunk, we need to get the boundary representation of human trunk including the external trunk skin and internal bone surface models. In this section, we briefly review the related 3D acquisition and reconstruction techniques. We use the Multiple X-ray Images technique to get the 3D internal bone surface model. The 3D external trunk skin surface model is acquired using Inspeck's 3D digital cameras.

1.1.1 3D Reconstruction of Internal Bone Model

Now many methods are available to construct the 3D model of the human skeleton. Non-invasive imaging techniques such as Magnetic Resonance Imaging (MRI) combined with 3D modeling techniques allow precise visualization of 3D bone structures. However, few hospitals can afford the MRI system. Efficient and affordable Computed Tomography (CT) systems can now be found in every hospital and many clinics. But for ethical reasons, hazardous X-ray can't be used extensively enough to provide full 3D data for large bone structures as the human trunk (Roy96). This is why for many years, clinicians have relied solely on 2D images to diagnose orthopedic problems and plan their treatment. Since X-ray images are 2D perspective projections, it is very difficult to appreciate the 3D deformation of the trunk just by looking at a frontal and lateral X-ray images of a scoliotic patient.

In their paper (Dansereau93; Delorme99), The researchers in Montreal presented an original multiple view X-ray images technique, which provides the human skeletal trunk from minimal personalized information (2D X-ray images) completed with high resolution generic 3D information (CT scan reconstructed anatomical primitives). The technique involves application and adaptation of known close-range photogrammetry techniques such as the DLT (McGlone89), as well as the development of a new 3D freeform deformation technique called 3D geometrical kriging (Krige51; Trochu93). The precision of the complete modeling process is evaluated and shows **point to surface** errors of $3.5 \pm 4.1mm$, which is acceptable for several applications such as diagnosis of spinal deformities and mechanical simulation of orthopedic treatments.

This 3D Multiple X-ray images modeling technique requires several steps:

1. Computer assisted X-ray image analysis;
2. 3D inference of anatomical landmarks position;
3. Anatomical primitive free-form deformation.

Three X-ray images are taken with special set-up (Figure 1.1): a posterior-anterior view (PA-0^o), a second posterior-anterior view with the X-ray tube lifted up and angled down at 20^o(PA-20^o), and a lateral view (LAT). Anatomical landmarks are manually identified on digital radiographic images and their 3D positions was inferred using the Direct Linear Transformation (DLT) method to solve the co-linearity conditions. 6 control points for each vertebra and 60 control points for each rib are extracted in this procedure (Figure 1.2). At last, anatomical primitive free-form deformation is performed based on these control points to reconstruct the 3D spine and rib cage model (Figure 1.3).

1.1.2 3D Reconstruction of External Trunk Surface Model

Two important factors should be considered in acquiring the 3D model of external trunk surface. The first is that we need a very precise model because it is a clinical application. The second is that the acquisition procedure should be as quick as possible because the flexible human body can easily change its shape in a short time.

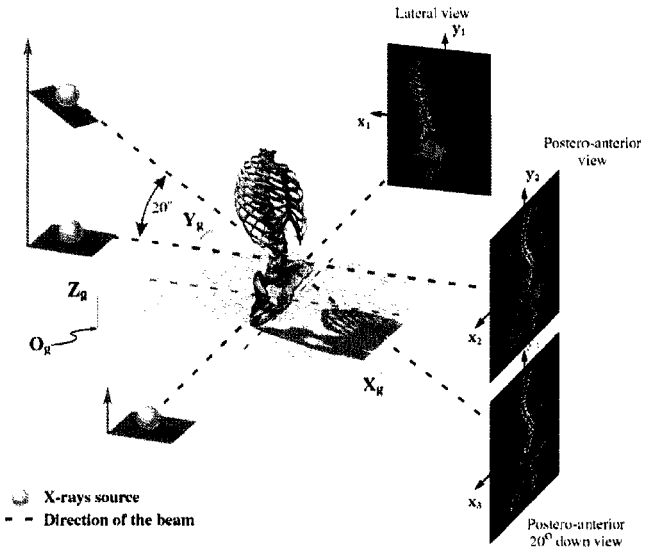


Figure 1.1 The set-up of Multiple X-ray images acquisition.

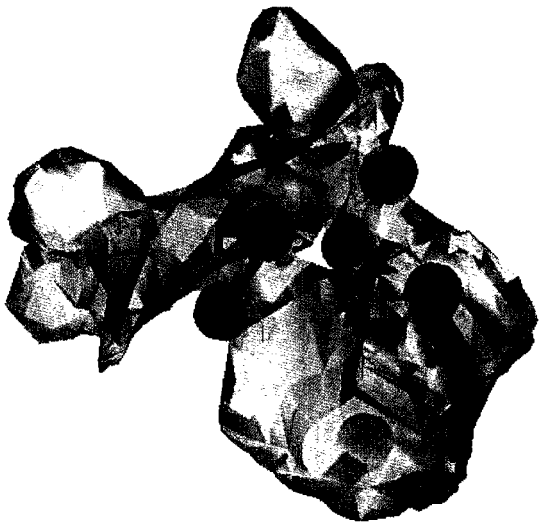


Figure 1.2 Six control points are extracted from multiple X-ray images for each vertebra according to the anatomical features.

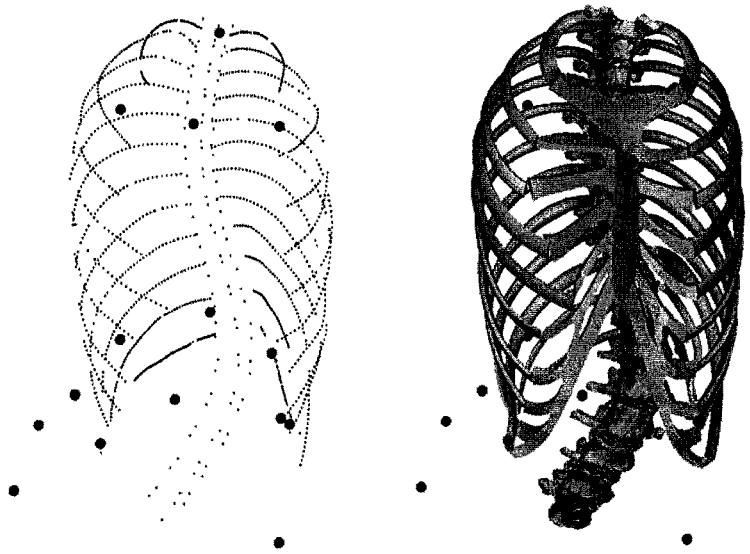


Figure 1.3 Reconstruction of the 3D spine and rib cage model from multiple X-ray images by anatomical primitive free-form deformation .

Several years ago, the research group in the University of Calgary used a laser scanning system (Poncet99) to capture the shape of the human trunk. The measurement error is less than $1.5mm$. But the whole scanning procedure needs about 10 seconds.

In Montreal, the researchers in Saint Justine Hospital use the 3D optical cameras from Inspeck (www.inspeck.com) to reconstruct the external trunk surface model. This 3D optical digitizer acquires range images that are processed by modeling software and converted to point positions in space. All of the surface visible by the optical sensors can therefore be digitized. The Inspeck cameras can obtain not only the shape of the surface, but also the color information (texture), which is automatically fitted to the surface. This is all done without contact and in a short time, which makes it well suited for the digitizing of flexible human trunk. Four cameras surrounding the patient at predefined optimal directions are used to collect four partial trunk models in less than 3 seconds (Figure 1.4). And these partial models are registered and merged into a whole trunk model by 3D modeling software **EM** (Editing and Merging, from Inspeck Company) (Figure 1.5). There are about 65,000 points, 130,000 triangles in this human trunk model. The average resolution is $1.1mm$.

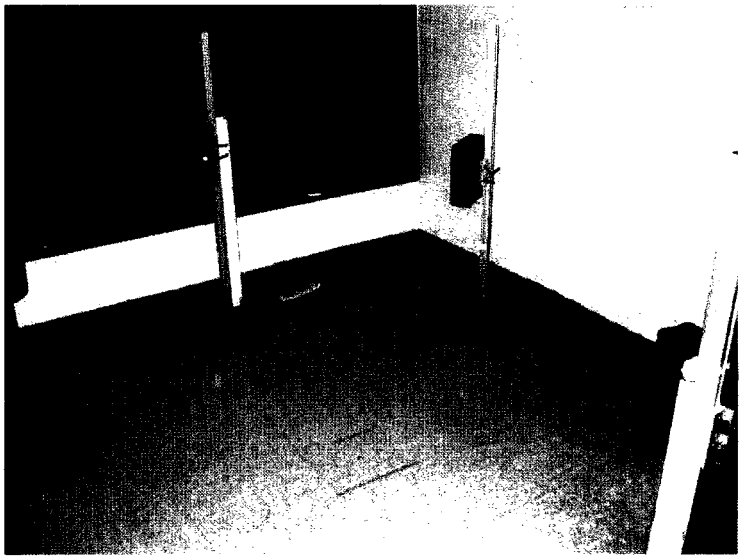


Figure 1.4 The set-up of the Inspeck's 3D cameras

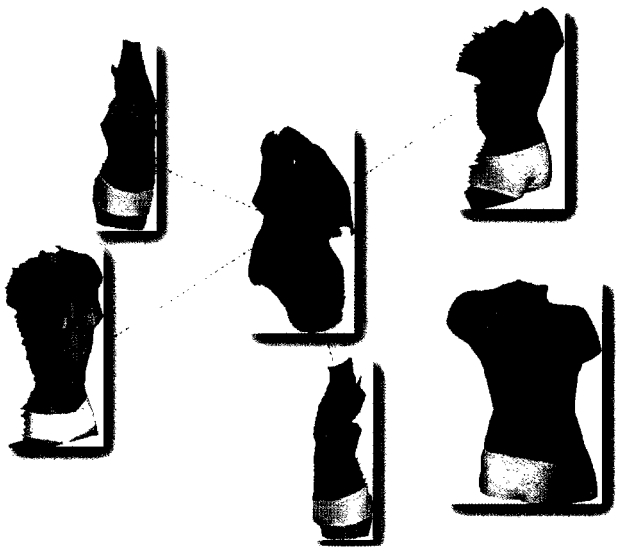


Figure 1.5 Merge different parts of model from the Inspeck's 3D cameras

In the next section we will review the registration techniques which are used to register and fuse these external trunk skin and internal bone surface models into the same coordinate system space.

1.2 Registration Techniques

The external trunk skin and internal bone surface models are acquired using different sensors at different times and sites. So a landmark-based registration procedure must be performed before fusing the two models together. The definition of the Landmark-based registration is as follows:

Definition 1 *Given a number n of corresponding point landmarks $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$, where $i = 1, \ldots, n$ in two image spaces of dimension d, find a* ***continuous*** *transformation $\boldsymbol{u} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ within a suitable Hilbert space $\mathcal{H}$ of admissible functions, which minimizes a given functional $\mathcal{J} : \mathcal{H} \rightarrow \mathbb{R}$ and fulfills the interpolation or approximation conditions $\boldsymbol{q}_i = \boldsymbol{u}(\boldsymbol{p}_i)$*

In the previous research (Fortin02), the rigid transform with least squares error metric has been used to register these two models into the same coordinate system space. The main purpose was to visualize the superimposed models to assist clinicians in the development of brace design. However, this global linear transform scheme is a coarse method because the trunk's shape will change when the patient moves or breathes during the acquisition period. It is necessary to take into account the elastic and local deformation.

In recent years, Thin-Plate Spline Interpolation / Approximation methods are popular for the elastic 2D or 3D multi-modal image registration. Bookstein (Bookstein89) proposed the use of Thin-Plate Spline interpolation for landmark-based registration and applied this scheme to 2D images. Application to 3D image data has been firstly reported by Evans et al (Evans91). And Rohr has developed the Thin-Plate Spline isotropic / anisotropic approximation schemes to register the 3D MRI and CT images of the human brain (Rohr01). However there is no report that these methods have been used to register the external trunk skin and internal bone surface models. In this project, we will evaluate the registration result of Thin-Plate Spline Interpolation / Approximation methods

on human trunk. Specifically, we need to determine optimal trade off parameter for the approximation scheme in the context of the human trunk.

In the following subsections, we will briefly introduce some basic theories of the rigid and elastic (Thin-Plate Spline) registration methods and other related algorithms.

1.2.1 Rigid and Affine Transformation

For all combinations of translation, isotropic scaling and rotation, we are looking for an equation of the following form:

$$\boldsymbol{u}(\boldsymbol{x}) = sR(\boldsymbol{x}) + \boldsymbol{t} \tag{1.1}$$

Where s is the scale factor. For the Rigid-Body transformation, the scale factor should be unit value. So $s = 1$. R is the rotating matrix, and $\boldsymbol{t}$ is the translating vector. And we use the **Least Sum of Squares Errors** as the metric.

$$\mathcal{J}(\boldsymbol{u}) = \sum_{i=1}^{n} \|\boldsymbol{e}_i\|^2 \quad \text{where} \quad \boldsymbol{e}_i = \boldsymbol{p}_i - \boldsymbol{u}(\boldsymbol{q}_i) \tag{1.2}$$

For the Affine transformation, the transform function is slightly different from the equation (1.1), It should be written as:

$$\boldsymbol{u}(\boldsymbol{x}) = X(\boldsymbol{x}) + \boldsymbol{t} \tag{1.3}$$

Where X is an arbitrary 3×3 matrix, t is the translating vector. The solution to these problems has been discussed in the paper (Horn87).

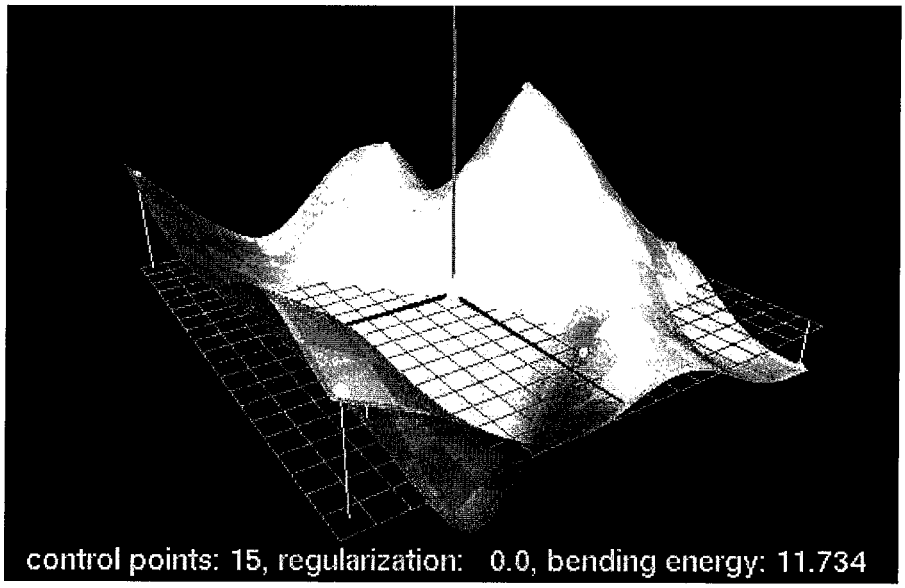


Figure 1.6 The thin plate will have the minimum bending energy in stable status (elonen.iki.fi).

1.2.2 Thin-Plate Spline Interpolation

A thin plate that is controlled by several points will have the minimum bending energy when it reaches the stable state (Figure 1.6). When the control points move, the shape of the thin plate will change also. But it always keeps the minimum bending energy when the change stops.

Thin-Plate Spline interpolation can be treated as a multivariate interpolation problem (Bookstein89). With this approach the minimizing functional represents the bending energy of a thin plate separately for each component $u_k, k = 1, \ldots, d$ of the transformation $\boldsymbol{u}$. Thus, the functional $J(\boldsymbol{u})$ can be separated into a sum of similar functions each of which only depends on one component u_k of $\boldsymbol{u}$. Therefore the problem of finding $\boldsymbol{u}$ can be decomposed into d problems.

In the case of d-dimensional images and for an arbitrary order m of derivatives in the functional we have

$$J_m^d(\boldsymbol{u}) = \sum_{k=1}^{d} J_m^d(u_k) \tag{1.4}$$

Where the single functional reads as

$$J_m^d(u) = \sum_{t_1+\ldots+t_d=m} \frac{m!}{t_1!\ldots t_d!} \times \int_{\mathbb{R}^d} \left(\frac{\partial^m u}{\partial x_1^{t_1}\ldots\partial x_d^{t_d}}\right)^2 d\boldsymbol{x} \tag{1.5}$$

The solution that minimizes the functional in equation 1.5 can be written in analytic form

$$u(\boldsymbol{x}) = \sum_{v=1}^{M} a_v \phi_v(\boldsymbol{x}) + \sum_{i=1}^{n} w_i U(\boldsymbol{x}, \boldsymbol{p}_i) \tag{1.6}$$

Where $M = (d+m-1)!/d!(m-1)!$. In the 3D space, $d = 3$, and we can let $\phi_1(\boldsymbol{x}) = 1, \phi_2(\boldsymbol{x}) = x, \phi_3(\boldsymbol{x}) = y, \phi_4(\boldsymbol{x}) = z$. And the basis functions are:

$$U(\boldsymbol{x}, \boldsymbol{p}) = \begin{cases} |\boldsymbol{x}-\boldsymbol{p}|^{2m-d} \ln|\boldsymbol{x}-\boldsymbol{p}|, & 2m-d \text{ is an even positive integer} \\ |\boldsymbol{x}-\boldsymbol{p}|^{2m-d}, & otherwise \end{cases} \tag{1.7}$$

For $m = 2, d = 3$ we have the simple distance function $U(\boldsymbol{x}, \boldsymbol{p}) = |\boldsymbol{x}-\boldsymbol{p}|$ (tps_{in}, kernel: r); For $m = 3, d = 3$, it should be $|\boldsymbol{x}-\boldsymbol{p}|^3$ ($tps_{in}(r^3)$, kernel: r^3), a smoother transformation than the transformation when $m = 2$. To compute the coefficients

$$\boldsymbol{a} = (a_1, \ldots, a_M)^T \quad \text{and} \quad \boldsymbol{w} = (w_1, \ldots, w_n)^T$$

of the analytic solution 1.6 we have to solve the following system of linear equations:

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{P} \\ \boldsymbol{P}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ 0 \end{bmatrix} \tag{1.8}$$

Where $\boldsymbol{K}_{ij} = U(\boldsymbol{p}_i, \boldsymbol{p}_j)$, $\boldsymbol{P}_{ij} = \phi_j(\boldsymbol{p}_i)$, and $\boldsymbol{v}$ is the column vector of one component of the coordinates of the landmarks $\boldsymbol{q}_i$. The condition $\boldsymbol{P}^T\boldsymbol{w} = 0$ represents the boundary conditions and ensures that the elastic part of the transformation is zero at infinity.

1.2.3 Thin-Plate Spline Approximation

With the interpolation approach described in the previous subsection, the landmarks are matched exactly. If we want to take into account landmark localization errors, we have to extend this approach by weakening the interpolation condition (Equation 1.4). This can be achieved by introducing a quadratic approximation term (Rohr01):

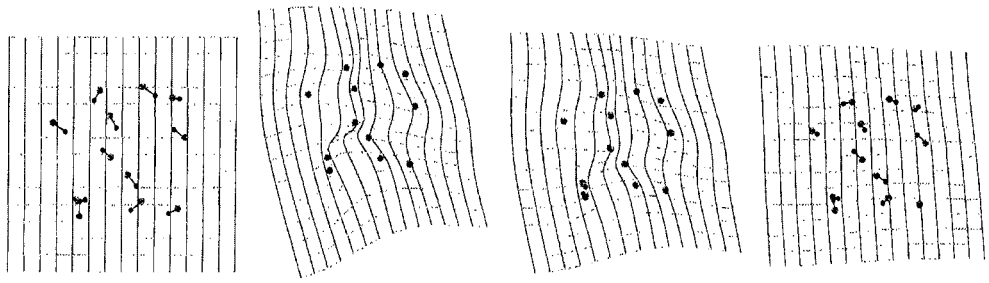
$$\boldsymbol{J}_\lambda(\boldsymbol{u}) = \frac{1}{n}\sum_{i=1}^{n}\frac{|\boldsymbol{q}_i - \boldsymbol{u}(\boldsymbol{p}_i)|^2}{\sigma_i^2} + \lambda \boldsymbol{J}_m^d(\boldsymbol{u}). \tag{1.9}$$

The first term measures the sum of the quadratic Euclidean distances between the transformed landmarks $\boldsymbol{p}_i$ and the landmarks $\boldsymbol{q}_i$. Each distance can be weighted by the variances σ_i^2 representing landmark localization errors. We use the sum of both variances, $\sigma_i^2 = \sigma_{i,p}^2 + \sigma_{i,q}^2$. The second term measures the smoothness of the resulting transformation.

The relative weight between the approximation behavior and the smoothness of the transformation is determined by the regularization parameter λ (see Figure 1.7).

- If λ is small, we obtain a solution which is similar with Thin-Plate Spline interpolation.
- If λ is large, we get a very smooth transformation which considers less local deformation.

The solution to the approximation problem (Equation 1.9) can also be stated analytically and consists of the same basis functions as in the case of interpolation. The computational scheme to compute the coefficients of the transformation u is nearly the same



(a) Original (b) $\lambda = 0$ (c) $\lambda = 0.001$ (d) $\lambda = 0.1$

Figure 1.7 Performance of approximating thin-plate splines visualized by deforming a regular grid (Rohr01).

$$\begin{aligned} (\boldsymbol{K} + n\lambda \boldsymbol{W}^{-1})\boldsymbol{w} + \boldsymbol{P}\boldsymbol{a} &= \boldsymbol{v} \\ \boldsymbol{P}^T\boldsymbol{w} &= \boldsymbol{0} \end{aligned} \tag{1.10}$$

where

$$\boldsymbol{W}^{-1} = \begin{pmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_n^2 \end{pmatrix} \tag{1.11}$$

1.2.4 Minimization Algorithm

To efficiently compute the optimal trade off parameter λ for the Thin-Plate Spline Approximation schemes (tps_{ap}), we need to use a minimization algorithm. In this section, we introduce the framework of minimization algorithms and two popular minimization algorithms.

Minimization algorithms begin with a bounded region known to contain a minimum. The region is described by a lower bound a and an upper bound b, with an estimate of the location of the minimum x.

The value of the function at x must be less than the value of the function at the ends of

the interval,

$$f(a) > f(x) < f(b)$$

Where a, b are the end points. This condition guarantees that a minimum is contained somewhere within the interval. On each iteration a new point x' is selected. If the new point is a better estimate of the minimum, $f(x') < f(x)$, then the current estimate of the minimum x is updated. The new point also allows the size of the bounded interval to be reduced, by choosing the most compact set of points which satisfies the constraint $f(a) > f(x) < f(b)$. The interval is reduced until it encloses the true minimum to a desired tolerance. This provides a best estimate of the location of the minimum and a rigorous error estimate.

Two popular minimization algorithms are:

Golden Section Algorithm The golden section algorithm is the simplest method of bracketing the minimum of a function. It is the slowest algorithm with linear convergence. On each iteration, the algorithm first compares the subintervals from the endpoints to the current minimum. The larger subinterval is divided in a golden section (using the famous ratio $(3 - \sqrt{5})/2 = 0.3189660\ldots$) and the value of the function at this new point is calculated.

The new value is used with the constraint $f(a') > f(x') < f(b')$ to select a new interval containing the minimum, by discarding the least useful point. This procedure can be continued indefinitely until the interval is sufficiently small. Choosing the golden section as the bisection ratio can be shown to provide the fastest convergence for this type of algorithm.

Brent Minimization algorithm The Brent Minimization algorithm combines parabolic interpolation with the golden section algorithm. This produces a fast algorithm which is still robust (Brent73). The outline of the algorithm can be summarized as follows: On each iteration, Brent's method approximates the function using an interpolating parabola through three existing points. The minimum of the parabola is taken as a guess for the minimum. If it lies within the bounds of the current interval then the interpolating point is accepted, and used to generate a smaller interval. If the interpolating point is not accepted then the algorithm falls back to

an ordinary golden section step. The full details of Brent's method include some additional checks to improve convergence.

The hole filling algorithms will be reviewed in the next section.

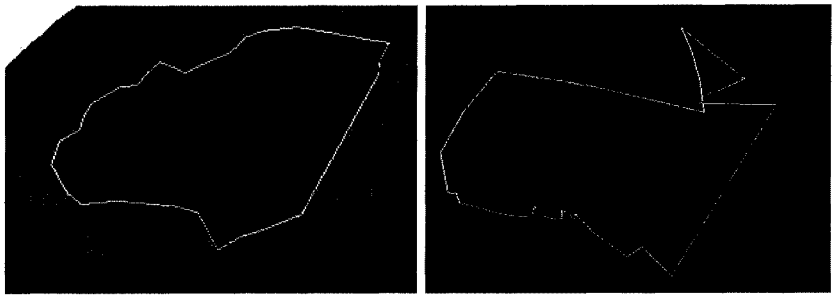
1.3 Surface Mesh Processing and Optimization Techniques

Inspeck 3D camera system can measure the shape of an object's surface with high accuracy and resolution. However, the optical cameras often can't observe the entire surface, so the resulting 3D models may be incomplete. In fact, all range finding systems have this kind of problems. On the other hand, when using the Inspeck 3D camera to capture the topography of a human trunk, the whole model is merged from several partial views. This operation is a complex procedure and easily produces errors at the overlapping regions. Another problem is that the arms, head, and legs are cut off from the trunk model. At these cutting places, the mesh quality is not good, and sometimes holes, non-manifold edges or self-intersecting triangles will appear.

In some applications, for visualization purposes, an incomplete surface model is enough — it represents the surface exactly as measured, without adding fabricated geometry. However, other applications require a watertight surface that bounds a volume of space. For 3D tetrahedron mesh generation, algorithms often require the surfaces to be valid 2-manifolds, closed, and without self-intersecting polygons.

As we know, the Finite Element Simulation is a heavy computation load task. In order to improve the computation performance, we need to reduce the number of elements. On the other hand, we should increase the resolution of the 3D tetrahedron mesh for improving simulation precision. In most 3D tetrahedron mesh generation algorithms, the resolution of the surface mesh bounding the volume determines the resolution of the 3D tetrahedron mesh of the volume. And the smoothness of the surface mesh also affects the quality and size of the 3D tetrahedron mesh.

To construct a watertight surface mesh, the holes in the mesh have to be filled. One difficulty of hole filling is choosing appropriate topology. Many holes are simple and can be filled with a disc topology (Figure 1.8(a)); In these cases, simple triangulation



(a) Simple hole (b) Complex hole

Figure 1.8 Different shape of the holes.

algorithms can be used. However, some holes have complex spatial structures (Figure 1.8(b)), simple triangulation algorithms can't handle them because naive triangulation of such holes often produce self-intersecting geometries.

In the following sections, we will introduce several hole filling algorithms:

- Simple Triangulation;
- Volumetric diffusion;
- Fast RBF (Radial Basic Function).

The first algorithm manipulates the edges and triangles directly. The last two methods are level set methods, and they will construct an implicit surface distance function, then reconstruct the closed surface by a marching cube or marching tetrahedron algorithm. We will put more emphasis on the FastRBF method.

1.3.1 Simple Triangulation

In general it is not possible to use a two dimensional algorithm to construct the triangulation for the hole in the 3D context, since the loop of the hole edges is usually non-planar.

For some simple flat holes, we can project the loop of the hole edges onto a plane, and the projection will form a non self-intersecting polygon. Then we triangulate this

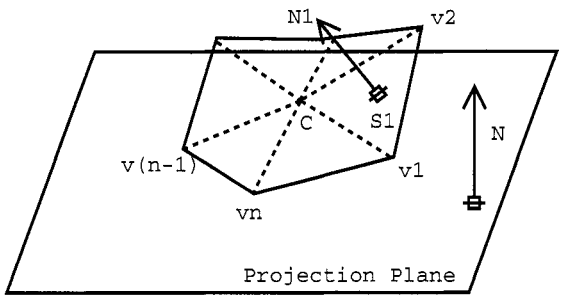


Figure 1.9 Finding the projection plane.

projection polygon, map the triangulation back to the original loop of the hole. And this triangulation method guarantees no self-intersecting geometry.

In all cases, finding the appropriate projection plane is not a trivial problem. A simple solution is to introduce a new point c, the centroid of the vertexes of the loop. Connect c with each edge to form a triangle loop. Then let the projection plane's normal to be the average of product of each triangle's normal direction $\vec{N}_i$ with its area s_i. See Figure 1.9, $v_1, \ldots, v_n$ are the vertexes of the hole edges; c is the center of the loop vertexes; $\vec{N}_1$ is the normal of $\triangle v_1 v_2 c$; and s_1 is the area of $\triangle v_1 v_2 c$.

$$c = \frac{1}{n}\sum_{i=1}^{n} v_i, \qquad \vec{N} = \sum_{i=1}^{n} \vec{N}_i s_i \tag{1.12}$$

However, this plane can't be guaranteed to be the projection plane without introducing self-intersecting edges for the projected polygons. And especially for some convoluted holes this projection plane doesn't exist.

A more aggressive method is based on this projection plane algorithm. Once we find that self-intersecting edges exist in the projected polygon of the projection plane, we will delete the corresponding triangle in the original mesh and then continue. The cost is that more original geometry information is lost. And in the extreme case,this algorithm will fail also.

A recursive loop splitting algorithm was reported by Schroeder (Schroeder92). See Figure 1.10, each time a splitting plane was found to split the hole's vertexes loop into two non-overlapping loops until only three vertexes remain in each loop. Of course, it is easy

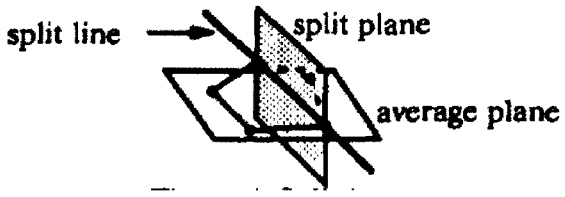


Figure 1.10 Recursive loop splitting triangulation method (Schroeder92).

to create examples where this algorithm will fail to produce a successful splitting plane.

Due to complexity of the geometry and topology of the holes, it is very hard to manipulate directly with the edge to produce a watertight surface mesh. So we give up the attempt to fill holes by this way.

1.3.2 Volumetric Diffusion

The researchers in Standford implemented an effective hole-filling algorithm — Volumetric Diffusion (Davis02).

First the original surface mesh is converted to the volumetric representation, which is a regularly spaced 3D voxel of values of a clamped signed distance functions $d_s(\boldsymbol{x})$. This function is defined only in a narrow band near the observed surface, and it is positive inside and negative outside. The observed surface is the zero set of d_s. The goal of this algorithm is to extend d_s to a function d that is defined over the entire volume, though in practice we only compute d near the surface—in fact only near holes in the surface. We achieve this by diffusing the values of d_s outward from the observed surface into the adjoining undefined areas. As the function spreads, so does its zero set. In particular, the diffused function propagates inward across the holes, eventually spanning them. Once the diffusion is complete, the zero set of this function is the desired hole-free surface. We can use the marching cube algorithm to get the closed polygonal mesh from this volume. Figure 1.11 illustrates an example of 2D diffusion in progress.

This volumetric diffusion algorithm, implemented naively, would consume time and memory proportional to $O(n^3)$, the number of voxel in the volume. Because most of the volumes are empty, and only a small fraction of the surface contains holes, this is

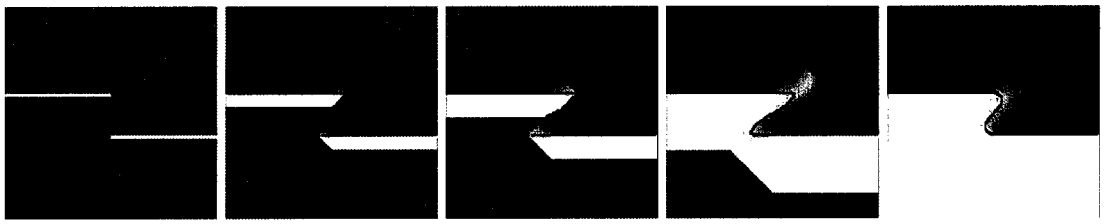


Figure 1.11 Illustration of 2D diffusion in progress. (Davis02)

inefficient for large models. Two measures could be taken to accelerate the computation:

- First, a sparse representation of the volume that avoids using memory for undefined areas;
- Second, we limit the computation to grid that are not more than a certain predetermined distance from a hole boundary in the original surface.

1.3.3 FastRBF Method

Recently the point-cloud methods was intensively studied. This method treats the union of all the scans as an unorganized set of 3D points to be fit with a continuous surface. With no connectivity between range samples, the large gap across a hole is conceptually equivalent to the space between adjacent samples, so these methods effectively fill holes during reconstruction.

A class of point-cloud methods fits a set of 3D Radial Basis Functions (RBFs) to the data; A weighted sum of these functions forms a new function, a level set of which is the intended surface (Dinh01) (Carr01). RBFs are popular for interpolating scattered data since the associated system of linear equations is guaranteed to be invertible under very mild conditions on the location of the data points. Radial Basis Functions do not require that the data lie on any sort of regular grid. In the Paper (Franke82), Franke identifies Radial Basis Functions as "often giving the most accurate results of all tested methods" and producing surfaces that are "usually pleasing and very smooth".

Widespread adoption of RBFs for data interpolation has been delayed because of their apparent extreme computational cost. Conventional methods for fitting an RBF by interpolation require $O(N^3)$ arithmetic operations, where N is the number of data points. This makes computation when N exceeds $10,000$ impractical even on mainframe systems. However, recently the Hierarchical Fast Multipole Methods (FMM) is emerging, which reduces the computational costs to $O(kN \log N), k \ll N$ combined with the preconditioned GMRES iteration (Beatson99). This makes the RBFs a practical method for large scale point sets. Once the implicit signed surface distance function of the original mesh was established, a marching tetrahedron variant was used to polygonize the surfaces mesh (Treece99).

The functional nature of the RBF representation offers new possibilities for surface registration algorithms, mesh simplification, compression and smoothing algorithms. A powerful toolbox has been implemented by the FarField company (Appendix III.3), which provides the above functions, and allows us to fill holes, and smooth and simplify the mesh at the same time.

In the following subsections, we will describe the FastRBF algorithm.

1.3.3.1 RBF Description and Terminology

A Radial Basis Function is a function of the form

$$s(\boldsymbol{x}) = p(\boldsymbol{x}) + \sum_{i=1}^{N} \lambda_i \Phi(\boldsymbol{x} - \boldsymbol{x}_i), \tag{1.13}$$

where:

- s is the *radial basis function.*
- p is a low degree polynomial, typically linear or quadratic,
- the λ_i's are the RBF coefficients,
- Φ is a real valued function called the basis function $(r, r^2 log(r), r^3, \ldots)$,

- the $\boldsymbol{x}_i$'s are the RBF centers.

Given a set of N points $\boldsymbol{x}_i$ and values f_i, the process of finding an *interpolating* RBF, s, such that,

$$s(\boldsymbol{x}_i) = f_i, \qquad i = 1, 2, \ldots, N \tag{1.14}$$

is called *fitting*. The fitted RBF is defined by λ_i, the coefficients of the basis functions in the summation, together with the coefficients of the polynomial term $p(\boldsymbol{x})$. If we let $p_1, \ldots, p_l$ be a monomial basis for polynomials of the degree of p, and $\boldsymbol{c} = (c_1, \ldots, c_l)$ be the coefficients that give $p(\boldsymbol{x})$ in terms of this basis, then the interpolation conditions (1.14), can be rewritten in matrix form as a linear system:

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{c} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} \tag{1.15}$$

where

$$\begin{aligned} A_{i,j} &= \phi(|\boldsymbol{x}_i - \boldsymbol{x}_j|), \qquad i, j = 1, \ldots, N, \\ P_{i,j} &= p_j(\boldsymbol{x}_i), \qquad i = 1, \ldots, N, \quad j = 1, \ldots, l. \end{aligned}$$

Solving the linear system equation (1.15) determines $\boldsymbol{\lambda}$ and $\boldsymbol{c}$, and hence $s(\boldsymbol{x})$. In fact, this equation is the same as the Thin-Plate Spline equation 1.8. They are just two different views of the same problem. And they both are the special case of the Krigging algorithm (a type of statistical method used to predict unknown points based on the known measurements using the regionalized variable theory).

A fast approximation method (see next subsection) was used to fit RBFs and to evaluate RBFs. This means that true equality at the interpolation nodes is never achieved. The maximum difference between the fitted RBF value and the given values,

$$\max_{i=1,\ldots,N} |s(\boldsymbol{x}_i) - f_i|,$$

is called the *fitting accuracy*. Given a point $\boldsymbol{x}$, calculating the value $s(\boldsymbol{x})$ is called *evaluating* the RBF. Once again, since we use fast approximation methods, the true RBF

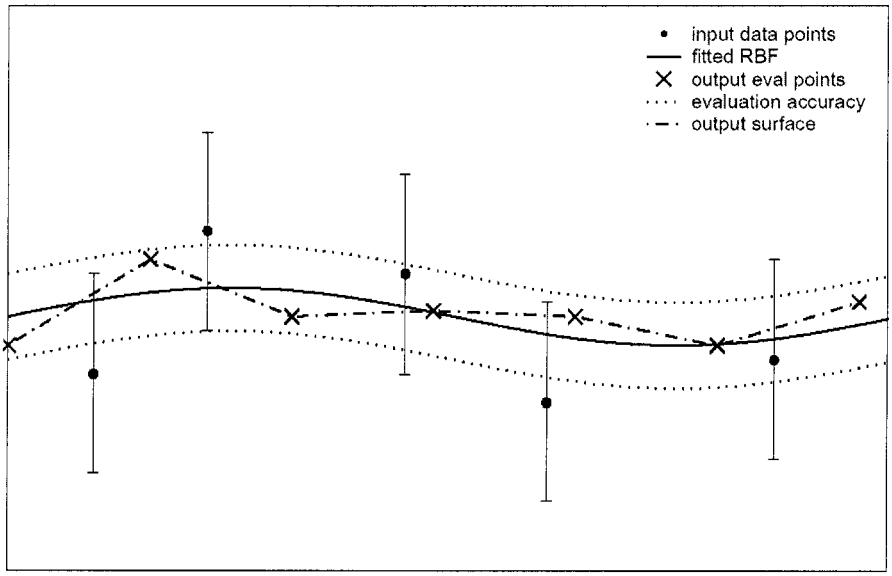


Figure 1.12 RBFs Fitter accuracy and evaluation accuracy (Farfield03).

value is not usually achieved. If a_i are the approximate values of the RBFs at the point $\boldsymbol{x}_i$, then the *evaluation accuracy* is the value

$$\max_{i=1,\ldots,N} |s(\boldsymbol{x}_i) - a_i|$$

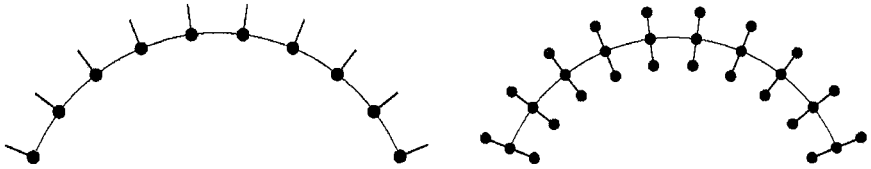
Generally, the evaluation accuracy should be much higher than the fitting accuracy. Figure 1.12 illustrates an exaggerated case of "interpolating" 1D data with RBFs.

To construct the implicit signed surface distance function for the incomplete mesh, we must construct some extra off-surface points with non-zero scalar value (Figure 1.13). The purpose of this procedure is to avoid the simple solution—zero everywhere.

1.3.3.2 Fast Multipole Method

For solving the large scale RBFs Interpolation / Approximation problems, we must solve the linear system equations

$$Mx = s$$



(a) Points on original mesh with value 0 and its normal direction.

(b) Construct new points along the normal direction, with non-zero values

Figure 1.13 Construct extra off surface point with nonzero value.(Farfield03).

where M is a $N \times N$ matrix, and x, s are vectors of length N. The direct solution, for example, Gauss elimination, LU Decomposition, SVD and etc., all require $O(N^2)$ storage and $O(n^3)$ operations, when N is large, the cost increases rapidly and becomes impossible even for mainframe systems.

However, we can use iterative methods to solve the RBFs Approximation problem. The iterative methods typically converge in k steps with each step needing a matrix vector multiply which normally require $O(N^2)$ operations. If properly designed, we can reach $k << N$.

The Fast Multipole Method (FMM) introduced by Rokhlin & Greengard in 1987, was called one of the 10 significant advances in computing of the 20^{th} century. It is an algorithm for achieving fast products of particular dense matrices with vectors in $O(N \log N)$ operations or even in $O(N)$. The key idea of FMM is to change the order of summation provided that the matrix entries are derived from particular functions that can be factorized or have their variables separated.

By combining the iterative methods (GMRES) and Fast Multipole Methods together, we can solve the RBFs linear equation in $O(kN \log N)$ time or even in $O(kN)$ time (Beatson99).

1.3.3.3 Surface Repolygonization

An RBF fitted to a set of surface data implicitly forms a solid model of an object. The surface of the object is the locus of points where the RBF is zero. This surface can be visualized directly using an implicit ray-tracer (Bloomenthal97), or an intermediate explicit representation, such as a mesh polygons, can be extracted. In the latter case, well known iso-surfacing algorithms such as Marching Cubes (Lorensen87) can be used to polygonize the surface. However, conventional implementations are optimized for visualizing a complete volume of data sampled on a regular voxel grid. The cost associated with evaluating an RBF means that an efficient surface-following algorithm is desirable.

In the implementation of FastRBF (Farfield03), a marching tetrahedron variant, optimized for surface following, has been used to polygonize surfaces. The surface following strategy computes the tangent at the known seed points on the surface and finds the next seed points on the surface in the tangent direction, thus avoiding the conventional requirement for a 3D array of sample points and therefore minimizing the number of RBF evaluations. Consequently, the computational cost increases with the square of the resolution, rather than the cube, as it would if a complete volume were sampled.

In the next section we will make a brief survey of the 3D mesh generation techniques.

1.4 3D Tetrahedron Mesh Generation

In the last decade, unstructured 3D mesh generation has developed rapidly. Many open source softwares are available and could be downloaded and used freely. We will investigate this field and try to find the most suitable tool for our purpose.

In this section, we will firstly give a survey of the fundamental algorithms in the mesh generation technology. Then an open source package— NETGEN— is introduced.

1.4.1 Structured vs. Unstructured Mesh

A structured mesh can be recognized by all interior nodes of the mesh having an equal number of adjacent elements. The basic elements typically are quadrilaterals (2D) or hexahedrons (3D). Algorithms employed generally involve complex iterative smoothing techniques that attempt to align elements with boundaries or physical domains. Structured meshes are most commonly used in the CFD field, where strict alignment of elements can be required by the analysis code or necessary to capture physical phenomenon.

Unstructured mesh generation, on the other hand, relaxes the node valence requirement, allowing any number of elements to meet at a single node. Triangles and Tetrahedral meshes are most commonly thought of when referring to unstructured meshes. For the analysis of the deformation of complex objects, this kind of mesh is more suitable and easier to generate than structured meshes.

In the following subsections, we will discuss the unstructured tetrahedral meshing techniques.

1.4.2 Octree Technique

The Octree was primarily developed in the 1980s by Mark Shephard's group (Yerry84) (Shephard91). Cubes containing the geometric model are recursively subdivided until the desired resolution is reached. Irregular cells are then created where cubes intersect the surface. Tetrahedrons are generated from both the irregular cells on the boundary and the internal regular cells (Figure 1.14).

The octree technique doesn't match the pre-defined surface mesh, as an advancing algorithm or Delaunay mesh might. The resulting mesh also will change as the orientation of the cubes is changed.

1.4.3 Delaunay Meshing

Many popular triangle and tetrahedral mesh generation algorithms are based on the Delaunay criterion (Delaunay34). The Delaunay criterion, sometimes called the "empty

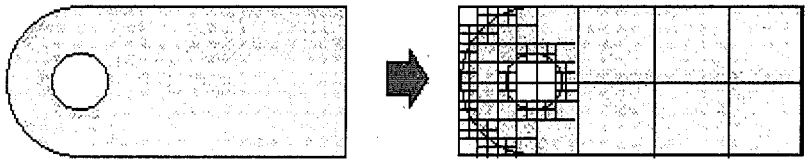


Figure 1.14 Quadtree decomposition of a simple 2D object (Owen03).

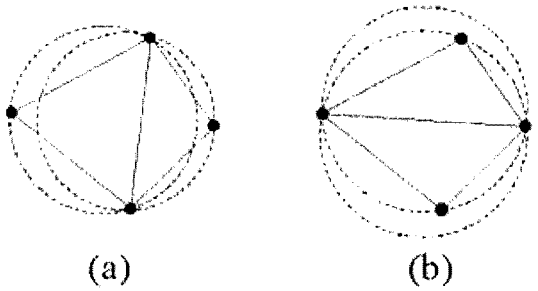


Figure 1.15 Example of Delaunay criterion. (a) maintains the criterion while (b) does not (Owen03).

sphere" property, says that any node must not be contained within the circumsphere of any tetrahedron within the mesh (Figure 1.15).

The Delaunay criterion in itself, is not an algorithm for generating a mesh. It merely provides the criterion by which to connect a set of existing points in space (Lawson77; Watson81). As such it is necessary to provide a method for generating node locations within the geometry. A typical approach is to first mesh the boundary of the geometry to provide an initial set of nodes. The boundary nodes are then triangulated according to the Delaunay criterion. Nodes are then inserted incrementally into the existing mesh, redefining the triangles or tetrahedral locally as each new node is inserted to maintain the criterion.

In many finite element applications, there is a requirement that an existing surface triangulation be maintained. This is called the *Boundary Constrained Triangulation.* Many methods have been proposed to recover the boundary (Weatherill94; George91; Joe91).

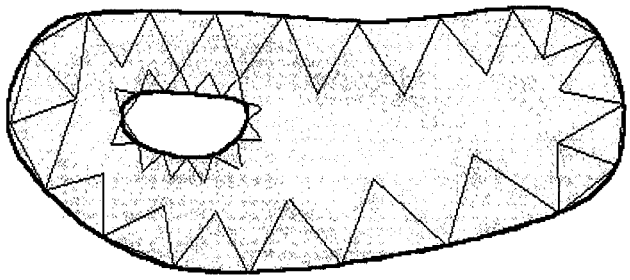


Figure 1.16 Example of advancing front where one layer of triangles has been placed (Owen03).

1.4.4 Advancing Front

Another very popular family of triangle and tetrahedral mesh generation algorithms is the advancing front method (Lohner88; Lohner96; Lo91a; Lo91b).

In this method, the tetrahedrons are built progressively inward from the triangulated surface. An active front is maintained where new tetrahedrons are formed (Figure 1.16). As the algorithm progresses, the front will advance to fill the remainder of the area.

1.4.5 NETGEN

Today, many robust, powerful commercial 3D mesh generation softwares are available and capable of generating unstructured tetrahedral meshes in multiple domain volumes. Anyway, we prefer the free and open source alternatives. In fact, most of the tools used in our project are from the open source community on the Linux platform. NetGen is one of them.

NetGen is an automatic mesh generation tool for two and three dimensions. NetGen is open source distributed under the the LGPL license. (Figure 1.17).

NetGen is very robust and many parameters could be adjusted to produce the desired high quality tetrahedral meshes. NetGen follows a top down strategy (Schoberl04). And we can control NetGen to run step by step according to the following list:

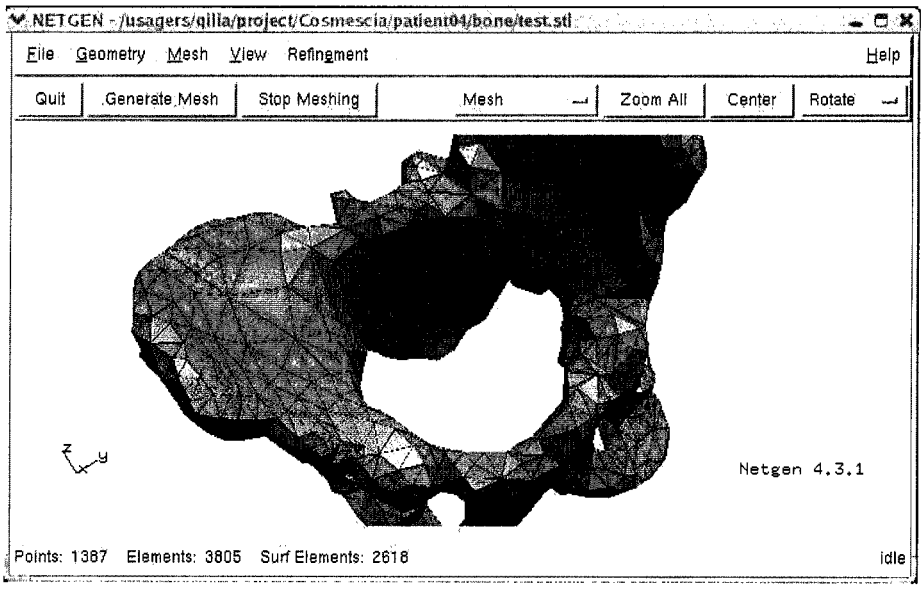


Figure 1.17 GUI of the open source mesh generation software — NETGEN.

Analyze geometry: It starts from computing the corner points (CSG format only);

Mesh edges: Then, the edges are defined and meshed into segments;

Mesh surfaces: The faces are meshed by an advancing front surface mesh generator;

Optimize surfaces: After meshing, the face meshes are optimized;

Mesh volume: The individual sub-domains are filled with tetrahedrons. Therefore, a fast Delaunay algorithm generates most of the elements (about 98%). But often it fails to mesh the whole domain, then the slower back-tracking rule-based algorithm takes over.

Optimize volume: Finally, the volume is optimized by the usual node movement, element swapping and splitting algorithms.

Unfortunately, NetGen doesn't support multiple domain tetrahedral mesh generation. We need to design a multiple domain mesh generation algorithm based on NetGen to generate a 3D mesh for the complex human trunk model.

1.5 Research Objective

The specific objective in this project is fusing the external and internal human trunk models by elastic deformation methods. And we also would like do some method investigation and development works towards the final objective, which is to fast and semi-automatically construct a precise personalized 3D tetrahedral mesh of the human trunk including the spinal column, rib cage, ligament, soft tissue and external trunk surface using the external trunk skin and internal bone surface models from Inspeck 3D cameras and Multiple X-ray images.

CHAPTER 2

MATERIAL AND METHOD

In this chapter, we will discuss the materials and methods that are necessary to fulfill our research objective. In section 2.1, we discuss the acquisition protocol of the external skin and internal bone surface models, and the extraction of the landmark positions. We design an evaluation scheme to compare the registration results of different methods in section 2.2. The external trunk shape prediction algorithm — is introduced in section 2.5. In section 2.3, we discuss how to preprocess the transformed external skin and the original interior bone surface model to get a simplified smooth and closed boundary representation of the multiple domain volume of a human trunk. In the last section, we propose the multiple domain 3D mesh generation algorithm based on NetGen to generate the 3D tetrahedral mesh of the human trunk.

2.1 Data Acquisition

2.1.1 Clinical Data Acquisition

All the external trunk skin and internal bone surface models are provided by Sainte-Justine Hospital (SJH) with the permission of the Ethical Committee of SJH, where all these data are routinely collected from the scoliotic patients for the purpose of diagnosis, brace design and surgery plan. As we have mentioned before, the 3D internal bone model are reconstructed from Multiple X-ray images, and the 3D external trunk skin surface models are acquired using Inspeck 3D cameras.

The postures which the patients hold for acquiring the external trunk skin and internal bone surface models should be carefully chosen. There is an important factor which should be considered: we need to guarantee the visibility of the anatomical structures (or whole external trunk shape) from the X-ray (or Inspeck optical cameras) which will affect the quality of the 3D reconstructed models. At present, the patients are asked to

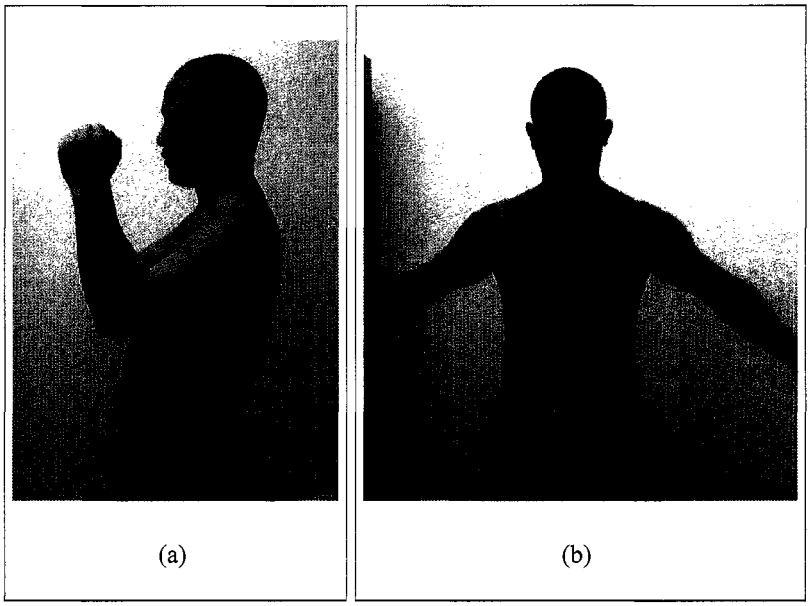


Figure 2.1 Different postures for acquiring the internal and external models. (a): optimal posture for the internal bone model using Multiple X-ray images; (b): optimal posture for the external trunk skin surface model using Inspeck 3D cameras.

hold the posture as in Figure 2.1(a) during the Multiple X-ray images acquisition for internal bone model, and to hold the posture like Figure 2.1(b) for acquiring external trunk surface models. The researchers in Sainte Justine Hospital are still studying the optimal postures for acquiring the internal and external models.

The researchers in Calgary used one single posture (hold the arms to both sides up to shoulder) for taking external and internal models. The reason why we don't adopt this schema is because we think that this non "nature" posture somehow changes the normal status of the human torso, especially of the scoliosis patients, and thus introduces some undesired affections to the following simulation work of the spinal deformity.

We intend to transform the external trunk model by landmark based elastic registration method into the coordinate system space where the internal bone surface model exists. Since the multiple X-rays technique has been well established and calibrated for a long time (compared with the Inspeck's 3D camera system) and many other related research works have been performed with the internal bone model as reference.

Since the external trunk skin surface model and internal bone surface model are acquired from two different postures, at different time, at different sites and by different sensors, we need to evaluate the precision of the fusion process. As we know, the fusion errors come from the following two factors:

1. the external and interior models are from different sensors;
2. the external and interior models are from different postures.

The multiple X-ray images technique and Inspeck's 3D camera system are well established and have relatively high precision. We can rely on theirs precision to certain degree. Obviously, the second factor introduces more deformation (error) than the first one and we need to study its effect more carefully.

We propose to evaluate the deformation effect by registering two different external trunk models of the same patient to make sure that the elastic methods are suitable for the flexible human trunk. For this evaluation purpose, we need several external trunk models of different posture for each patient. We will discuss this evaluation scheme in detail in section 2.2.

We take 4 external trunk models for each patient: 2 external models are from the posture as Figure 2.1(a), the others two models are from the postures as Figure 2.1(b). The naming protocol for these external models is "$\{F/G\}xxxx000i$", where F means the patient is female and G for male patient; $xxxx$ is the number of the patient; i is the index of the models, 0 and 1 mean the postures are like Figure 2.1(a), 2 and 3 like Figure 2.1(b) respectively.

We acquire only 1 internal bone surface model for each patient due to the risk of the X-ray to human body, especially for the adolescent patients. The positions of the radio-opaque markers on skin surface (see next subsection) and the anatomical feature points on each vertebrae (see subsection 1.1.1) are also extracted during the reconstruction procedure and provided together with this internal model.

All these external and internal models are taken in the same day. That means we needn't worry about the physical growth and progression of spinal deformity of the patient.

2.1.2 Landmark Distribution and Extraction

About 20 radio-opaque markers are put on the patient's skin surface for registration purpose. The location of these markers should be carefully chosen. At present, more markers are put on the back surface because they are easily identified via the posterior-anterior X-ray images (PA^0, PA^{20}) without being blocked by internal bone structure. Another reason is that the information of spinal deformity are more obvious in the back region of trunk than the front part.

However their distribution should be as sparse as possible in order to improve the registration precision. Since the trunk shape in the abdomen and chest region is most easily affected by the breathing and moving. The optimal distribution of the radio-opaque markers is still studied.

The coordinates of these markers in the 3D internal bone model are extracted during the reconstruction period from the Multiple X-ray images. Sainte-Justine Hospital has provided us these marker positions for each internal bone surface model.

We still need to get these markers coordinates in the coordinate system space where the external trunk skin surface model exists. As we know, the Inspeck 3D camera is capable of digitizing the trunk shape geometry and capturing the texture at the same time with high resolution. So we can locate the landmarks with the help of texture. We have implemented a GUI tool to render the external trunk model with texture and get the landmark coordinates interactively (Figure 2.2). The original external trunk model data file use the Inspeck's private file format (SZE) (Appendix II.1) which makes use of the texture atlas technique. Some conversion works have been done to extract the point coordinates, mesh indices and texture coordinates.

First, we render the external trunk skin model in parallel projection mode with texture mapping in the window of our GUI tool. Then we get the landmark coordinates (x_S, y_S) in the screen window by double clicking on markers position using texture information (Figure 2.2). Construct a ray:

$$R_S = O_S + DIR_S * t, \qquad t \geq 0. \tag{2.1}$$

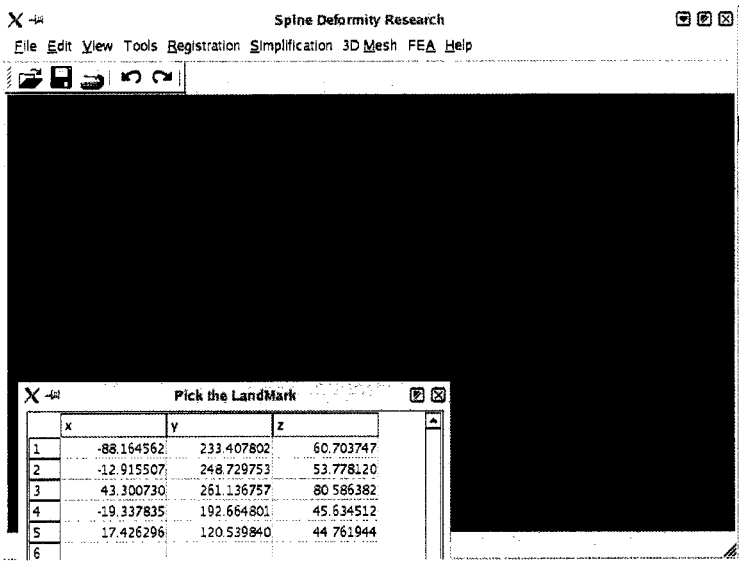
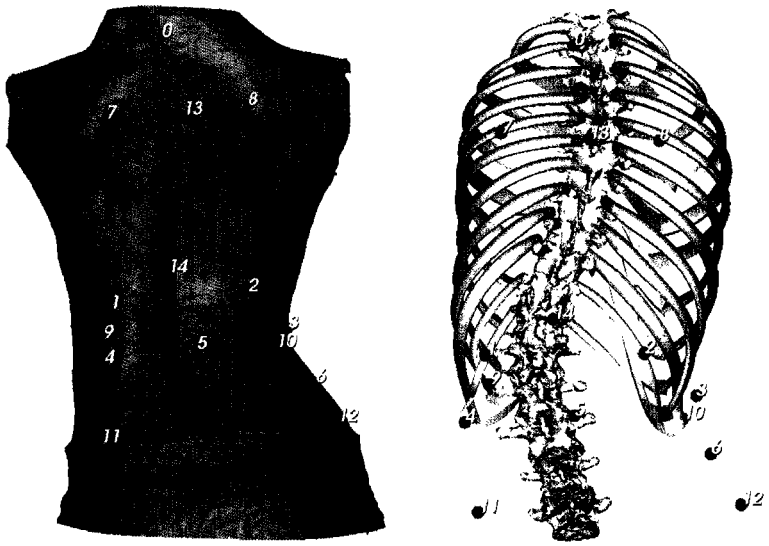


Figure 2.2 Pick the landmark's positions with the help of texture information on the trunk model.

Where $O_S = (x_S, y_S, 0)$, and $DIR_S = (0, 0, 1)$.

Then a reverse procedure of the parallel projection is applied to convert this ray from window screen coordinate space into the model coordinate space. Finally we intersect this ray with the trunk surface model. The first hit point is considered as the position of the landmark. As we know, there are about 65,000 points and 135,000 triangles in the trunk surface model. So a **Fast Spatial Search** technology needs to be used to accelerate this procedure. (Here we use the OCTREE data structure).

The number of markers which we can extract from the external and internal models and for which we can establish a correspondence relationship is often less than the number of markers which we have actually put on the patient skin surface. That is because sometimes some markers are blocked by internal bone structure and can't be identified from the multiple X-ray images. The Inspeck's merging procedure sometimes will also lose some texture information in the merging regions. Usually the landmarks that can be used are around 14 to 18 out of the original 20. Figure 2.3 shows an example of the external trunk skin and internal bone surface models with corresponding landmarks.



(a) (b)

Figure 2.3 The external trunk skin and internal bone surface models with the corresponding landmark sets ($F00080000$). Some markers in the front of the trunk are not visible.

2.2 Registration and Fusion Procedure

As we have mentioned in the previous section, we propose to evaluate the transformation results of different registration methods through two different external trunk skin surface models. We will introduce the evaluation procedure and the objective metric in subsection 2.2.1 and 2.2.2 respectively. In subsection 3.2.1, we present the scheme to compute the optimal trade off parameter of the Thin-Plate Spline Approximation (tps_{ap}). In the last subsection, we discuss how to directly evaluate the fusion results of the external and internal models.

2.2.1 Evaluation Procedure

Since it is hard to estimate the overall registration error, and the multiple X-ray images technique and Inspeck 3D camera system have relatively high precision, We propose to just evaluate the registration error introduced by the difference of postures through two different external trunk skin surface models. The procedure is as follows:

1. First choose one from the 4 available external models as **source** model, which posture is like Figure 2.1(b);
2. Choose another external trunk model as **target** model, which posture is like Figure 2.1(a) — same posture for the internal bone model; However, sometimes we also want to evaluate the transformation result between two different external trunk models from the same posture.
3. Then we extract the source landmarks $\{M_{src}\}$ and target landmarks $\{M_{tar}\}$ from the source model and target models respectively, and establish the corresponding relationship between these two sets of landmarks;
4. Transform the source model by different registration methods,
5. Finally we compare the transformed source models with the target model according to an objective **metric**. We will introduce this metric in more detail in the next subsection.

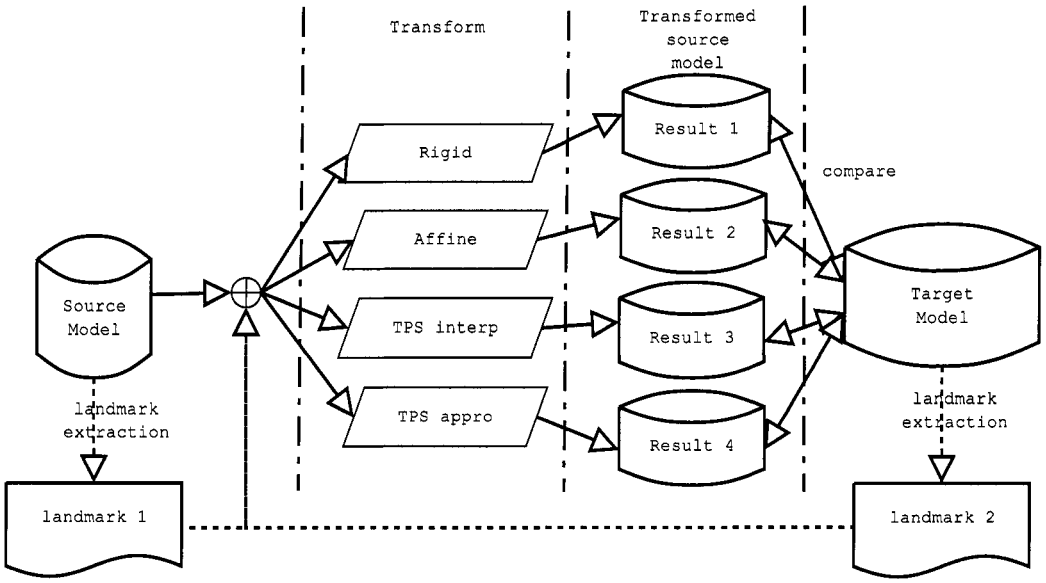


Figure 2.4 Evaluating and comparing the transformation results of the different registration methods.

Figure 2.4 illustrates the evaluation procedure.

2.2.2 Evaluation Metric

For comparing the transformed source external trunk model and target external trunk model, an objective metric must be established. Two methods are available.

The first one is **Volumetric Ratio** (Delorme99). The transformed source model and the target model are converted into volumetric representation, and the XOR/OR voxel operations are used to evaluate the ratio of the non-intersected (XOR) volume on the union (OR) volume of the two models. This method gives a value between 0% and 100%: 0% if there is a perfect match and 100% if the two trunk models have absolutely no voxel in common. The volumetric method is a global index. It can indicate the ratio of the difference of two models. However it can't tell distribution of the error. In our case, we hope to see how the different regions on the trunk surface are affected by different registration method.

We use the **Bidirectional Point-To-Surface Mean Distance** ($BPTSMD$) metric. For

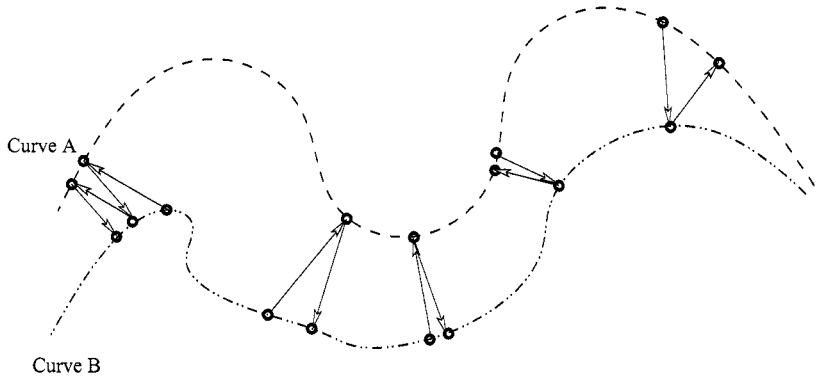


Figure 2.5 Illustration of the bidirectional point-to-curve minimum distance. (red lines are minimum distances from the points on curve A to curve B; green lines are the minimum distances from the points on curve B to curve A). The situation of $BPTSMD$ in 3D context is similar with this simple 2D example.

each point on the transformed source model, we calculate its minimum distance $d_i^{s\to t}$ to the surface of the target model; For each point on the target model, we calculate its minimum distance $d_j^{t\to s}$ to the surface of the transformed source model; Then we compute the overall mean value for all these two sets of minimum distances. It is very important for us to know the mean value of $\{d_i^{s\to t}\}$ is not equal to the mean value of $\{d_j^{t\to s}\}$, that is why we need to use the bidirectional mean distance (see Figure 2.5).

We know the external trunk models from the Inspeck's 3D camera system are large scale surface meshes (about 65,000 points, and 130,000 triangles per model), and the vertexes are densely and uniformly distributed. In the limit case (very dense vertex distribution), this BPTSMD value will be similar with the volumetric ratio (normalized by the overall surface area instead of overall volume).

This $BPTSMD$ metric is easy to implement. However, there is still a problem due to the large number of points and polygons. It will take a long time to compute if we implement it in the naïve way. An acceleration method based on OCTREE spatial search algorithm was used again to speed up the computation without sacrificing the precision. The acceleration scheme is stated as follows:

1. First we generate an OCTREE subdivision for the transformed source and target models respectively, where each octant carries an indication of whether it is empty

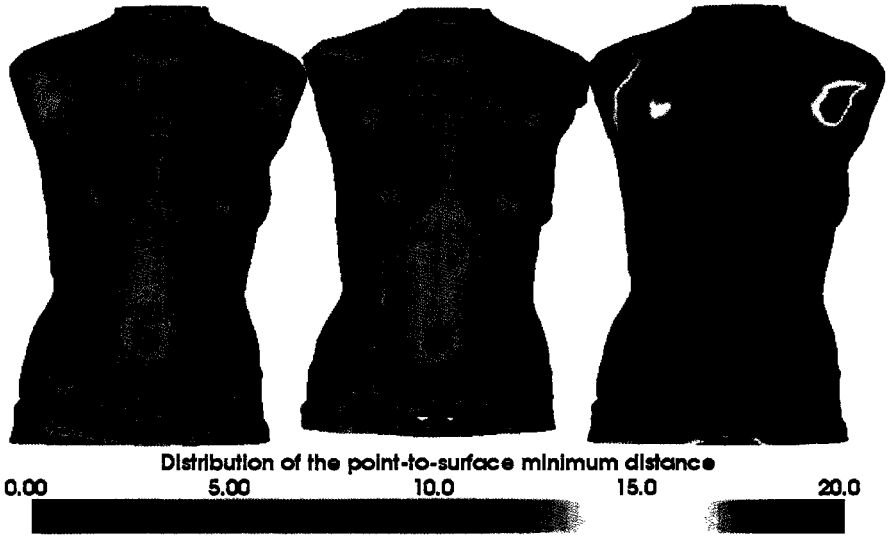


Figure 2.6 Distribution of the Point-to-Surface Minimum Distance. Left figure: transformed source model ($F00030002$); Middle figure: target model ($F00030000$); Right figure: the color indicates the minimum distance from the points on transformed source model to the surface of the target model. (Metric: mm)

or not (an octant is not empty if it has one or more triangles inside of it), and each non-empty leaf octant (finest level octant) carries a list of the triangles inside of it.

2. Then for each point in the transformed source model, we choose its nearest non-empty octant in the OCTREE subdivision of the target model, and compute the closest distance of the point with the triangles in these octants. This closest distance is the minimum distance of the point with the surface of the target trunk model. Here a fast and robust algorithm was used to compute the point-to-triangle distance (Eberly01).

3. Do the same thing for each point on target model to calculate its minimum distance to the surface of the transformed source model.

4. Compute the mean value for all these minimum distances, we get $BPTSMD$.

By this way, $BPTSMD$ can be calculated in less than 10 seconds using the machine in our lab (CPU: PIII 1.8G; Memory: 512M), which is acceptable for our application.

Another advantage of the $BPTSMD$ method is that we can easily visualize the registration error distribution on the trunk surface model. In Figure 2.6, the left model is the transformed source model ($F00030002$) constructed using Thin-Plate Spline Interpolation method. The middle model is the target model ($F00030000$). We compare these two models and compute the minimum distance from each point on the transformed source model to the surface of the target model. The result is showed in the right model. Notice the red and yellow colors in the part of the neck, shoulder and waist, that indicates that the errors in these regions are bigger than in others.

The external trunk model from Inspeck 3D camera includes the part of the human body which is below the neck and above the waist, and without the arms. These occluded parts are manually cut off at arbitrary positions. So the Point-to-Surface minimum distances for the points in these parts are larger than other places due to the non-consistency. It make the $BPTSMD$ metric unreliable. We should clip away the parts if they are not included in both models. However, this method is not easy to implement, because it needs manual identification.

We propose an efficient and effective method based on the fact that all the external trunk models are perpendicular to the xz-plane due to acquisition method by Inspeck 3D cameras. For the target model, we compute its highest and lowest landmarks (in y dimension). We construct two planes which parallel with the xz-plane. Only the points which are between these two planes will take part into the computation of $BPTSMD$. Figure 2.7 illustrates the region of interest between the two parallel planes (parallel with xz-plane).

We have to admit that by this way the non-consistency problem in the shoulder and arm region has not been dealt with. And we hope to get a more reasonable clipping scheme in future work.

2.2.3 Evaluation of Fusion Result

In previous subsections, we have discussed how to evaluate the registration error from the difference of posture through two external trunk skin surface models. But we still want to find an evaluation scheme to compare the fusion result of the external and internal

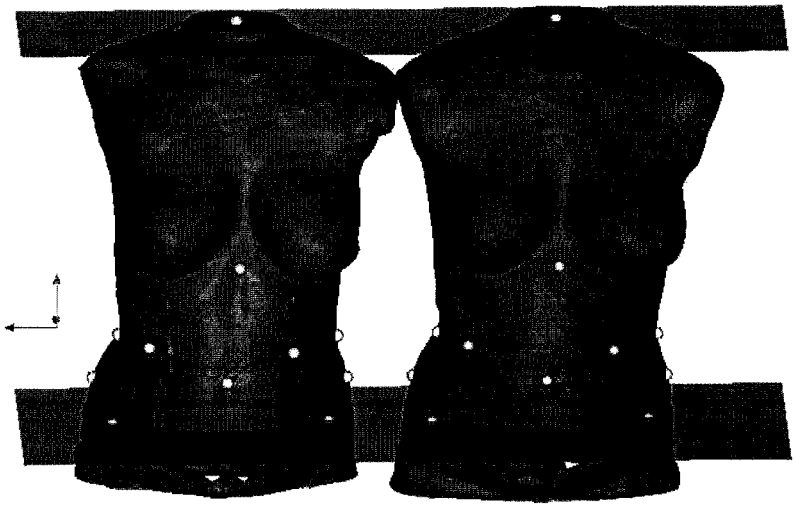


Figure 2.7 Construct the reliable region for computing $BPTSMD$. the interest region is between two parallel planes passing through the highest and lowest landmarks and parallel with the xz-plane. Left model is the transformed source model ($F00030002$), the right model is the target model ($F00030000$).

models that considers the registration error from the difference of sensors. Because what we really need to do at last is to register and fuse the external and internal models, instead of two external models. Unfortunately,

We noticed that after registration and fusion, the internal bone models often break out of the transformed external trunk skin surface (Figure 2.8). We intend to calculate the volume of the bone which break out of external trunk surface to see whether there exists any relationship between the outbreaking volume of bone and the registration methods. We need the following steps to accomplish this task:

1. First, we produce the watertight surface mesh for the external trunk and internal bone models. (We use FastRBF method to fulfill this task, and the detailed operations are discussed in the next sections).

2. We use different registration methods (rigid and elastic) to transform the watertight external trunk surface mesh.

3. Then the robust boolean set operation (using GTS, see Appendix III.5) on closed surface mesh is performed to compute the part of bone which breaks out of skin

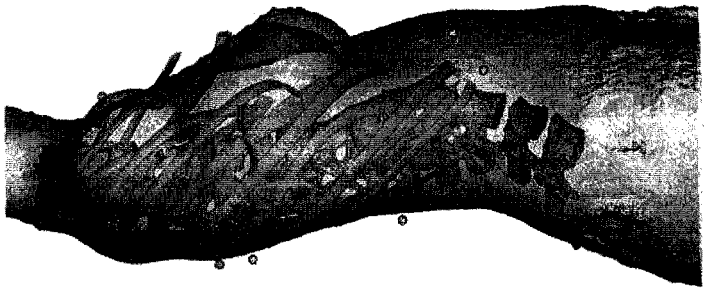


Figure 2.8 An example of registration and fusion result: rib break out of the trunk surface. (F00060000)

surface.

$$R_{outbreak} = R_{bone} \cap \overline{R_{trunk}}$$

4. Compute the volume of these parts of bone. The volume of the closed surface mesh could be computed using the following equation:

$$V_{total} = \sum_{i=1}^{n} V_{c,t_i}$$

Where c can be an arbitrary point; $\{t_i\}$ are the oriented triangles in the closed surface mesh; And V_{c,t_i} is the volume of the "oriented" tetrahedron formed by point c and triangle t_i. Note this volume could be negative.

We hope we could reveal more hidden problems by this way.

2.3 External and Internal Surface Model Preprocessing

All the 3D tetrahedron mesh generation algorithms require a valid volume model (CSG) or watertight boundary representation (B-rep) input that defines a closed space region which needs to be meshed. The situation is more complex when we need to handle two or more adjacent space regions, because we must take care of the adjacent boundary surface.

To reduce the computational load of the Finite Element simulation, the number of tetrahedron elements should be as small as possible. On the contrary, in order to improve the simulation precision, we must produce a finer volume mesh. So there is a trade off between the simulation precision and cost. A good surface mesh model for the external trunk skin and internal bone structure is very important to generate the 3D tetrahedral mesh which satisfy both requirements.

In this section, a series of algorithms and techniques are introduced to produce the watertight, smooth, precise and simplified surface mesh model for the external trunk and internal bone.

2.3.1 External Model Processing

Due to the availability of the toolbox of the FastRBF (Appendix III.3) and its hole filling, optimized repolygonization, simplification and smoothing abilities, we adopt it as our primary tool to process the external trunk skin surface model.

To use FastRBF to process the external trunk surface model, the first thing we should do is to remove the non-manifold edges (the edge connected to more than two triangles), which is not allowed in the context of valid boundary representation of volume. See Figure 2.9 for non-manifold edge examples. Due to the complexity of the surface mesh topology, there is no easy way to remove the non-manifold edges without introducing new holes.

A simple aggressive linear time $O(n)$ algorithm is used to fulfill this task. The basic idea is as the following:

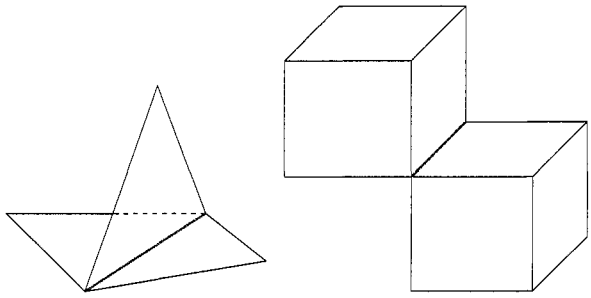


Figure 2.9 Examples of non-manifold edge. The wider lines are non-manifold edges which are connected to more than 2 faces.

1. Create a map with the edge as key. edge $e(v_1, v_2)$ is same as edge $e(v_2, v_1)$ (v_1, v_2 are the id's of the vertexes). Let each edge's reference count to be zero.

2. Iterate for each triangle t_i in the mesh, figure out the edges used by this triangle, and increase the reference count of these edges by 1. And record the reference relationship from edge to triangle at the same time.

3. Then check each entry in the edge map. All the edges which reference count are greater than 2 are the non-manifold edges.

4. Remove all the non-manifold edges, and also remove its associated triangles (it is an easy job because we have recorded the pointers from edge to triangle at step 2).

5. At last, we produce a valid surface mesh without non-manifold edges, but with more and bigger holes. In practice, this mesh often is composed by several separated parts, and most the points (more than 99.99%) are in the biggest one. We need to separate them and extract the biggest mesh part. The topology separation algorithm which will be discussed in subsection 2.3.3 can do this work.

Table 2.1 shows an example of an edge map. Three triangles connect with edge e_4, so it is a non-manifold edge.

This algorithm is also useful to find the boundary edges. The boundary edges are used by only one triangle. So all that we need to do is just to modify the step 3. In Table 2.1, edge e_3 is a boundary edge.

Table 2.1 Edge map used to classify the edges (example). e_4 is a non-manifold edge; e_3 is a boundary edge.

Edge	Vertexes	Reference count	Reference triangles
e_1	(v_1, v_2)	2	t_1, t_2
e_2	(v_2, v_4)	2	t_2, t_3
e_3	(v_3, v_1)	1	t_1
e_4	(v_3, v_4)	3	t_3, t_4, t_5
...	...	...	...

After the non-manifold edges are removed, we input this external trunk model into FastRBF to get a smooth, simplified, regularized and watertight surface model. Many parameters are available to control the processing quality, we need to adjust them carefully. The processing results are displayed in next chapter.

2.3.2 3D Interior Bone Model Investigation

In the previous subsection, we have proposed to use FastRBF to process the external trunk surface and produce the simplified smooth watertight mesh for the human torso. And we need do the same work for the internal vertebrae and rib cage bone structure also. But this time the situation is more complicated.

Since the 3D internal bone surface model provided by Sainte-Justine Hospital is mainly for the visualization and diagnosis purpose, we encounter many problems when we try to get a valid boundary representation (B-rep) of the internal bone volume from this surface model (Figure 2.10). In reality,

- Ribs shouldn't intersect with vertebrae;
- Ribs shouldn't intersect with each others;
- Vertebrae shouldn't intersect with each others;
- Vertebrae and rib cage shouldn't break out of the trunk surface.

However, at present we have to limit ourselves to start with this already provided model

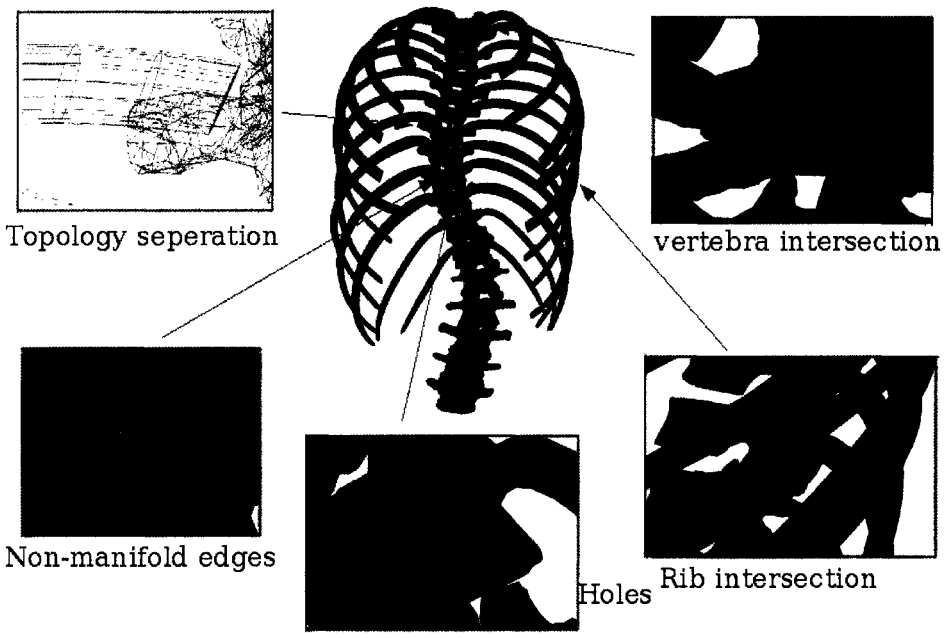


Figure 2.10 The situation of the 3D human skeletal trunk provided by Sainte-Justine Hospital (not include pelvis and sacrum).

to be sure that we can get a preliminary result in limited time. So in following subsections, we will talk about how to handle all kinds of bad situations to construct the valid surface mesh of spine, rib cage, pelvis, and sacrum. Many operations were taken to simplify and accelerate this procedure without obeying to the actual anatomy structure. But we believe some of our ideas and methods will still be also useful in future work.

2.3.3 Topology Separation

The 3D internal bone surface model from Sainte-Justine Hospital is in a single file, all the parts are located in the single model. To manipulate each part of bone directly, we need to separate them first.

Although geometric intersections occur, each part of bone (rib, vertebra, pelvis, sacrum) are topologically disconnected . We can extract each bone according to the mesh connectivity (Figure 2.11). A simple **Traverse and Mark** algorithm is able to split the whole model into parts. The algorithm is stated as the following:

1. Loop over edges, record the pointer from vertex to edge into a list which belongs to the vertex. And set all vertex's visited flag to be $false$ (Table 2.2 and 2.3). — $O(n)$ time.
2. Start from any not visited vertex v,
3. Set the visited flag of v to be $true$.
4. Iterate over the connected edge list of v, find a not visited vertex v' which is connected to v by the edge.
5. Recursively do the step 3 for the vertex v', until no more vertex is able to be found by this way. Then one disconnected part of the mesh was extracted.
6. Then we continue loop at step 2, until all vertexes were visited.
7. Since all the vertexes are only visited one time, the whole loop and recursion complexity is linear. — $O(n)$ time.

Altogether, we can separate the whole internal bone model into parts in linear time.

Table 2.2 Edge data structure (example).

Edge	Vertex 1	Vertex 2
e_1	v_1	v_2
e_2	v_2	v_4
e_3	v_3	v_1
e_4	v_3	v_4
$\cdots$	$\cdots$	$\cdots$

Table 2.3 Vertex data structure (example).

Vertex	Connected edges	Visited Flag
v_1	e_1, e_2, e_7, e_6	$true$
v_2	e_2, e_4, e_9	$true$
v_3	e_3, $\cdots$	$false$
v_4	e_4, $\cdots$	$false$
$\cdots$	$\cdots$	$\cdots$

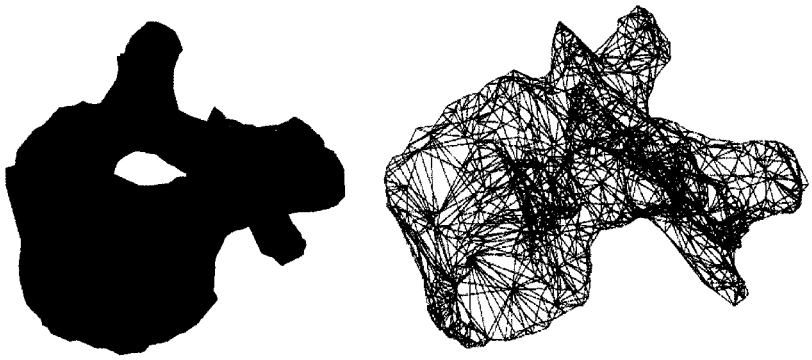
2.3.4 Vertebra Model Processing

There are holes and non-manifold edges in the original vertebra model (Figure 2.12). So once again, we use our non-manifold edge removing algorithm and FastRBF tool to process the vertebra and produce a watertight and regularized model.

In order to decrease the computational load of the Finite Element simulation, we should reduce the number of the basic elements as much as possible. We know that the vertebra is a rigid object compared with the soft tissue. We can use fewer tetrahedrons to represent it without sacrificing too much simulation precision. However, due to complex 3D shape of the vertebra, this time we can't use FastRBF to simplify the vertebra model.

We use the interactive 3D modeling tool — *Wings3D* (Appendix III.7) — to simplify the vertebra model manually. This step is necessary and fruitful because it can reduce dramatically the number of points, and thus reduce the number of tetrahedrons. Considering we have 17 vertebrae, it is hard and tedious work. But this work will be rewarded for the afterward use. Once we produce a set of simplified vertebra model, we can deform them to fit others patients — an approach to automatically personalized FE mesh generation.

Figure 2.11 Split the 3D human skeletal trunk model into parts.



(a) Original vertebra model (b) Wireframe of the original model

Figure 2.12 Original vertebra model with holes and non-manifold edges.

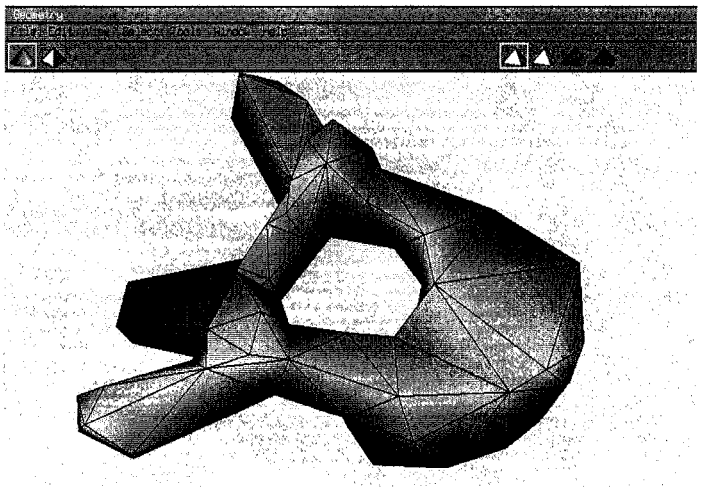


Figure 2.13 Simplify the vertebra model by hand (Using the free and open source software: *Wings3D*. www.wings3d.com).

Table 2.4 Deformation relationship for each vertebra.

Simplified Model	Deformation target
T4	T1, T2, T3, T4, T5, T6
T9	T7, T8, T9, T10, T11
L2	T12, L1, L2, L3, L4, L5

We know the vertebrae which are close with each other nearly have the same shape. We choose to simplify several key vertebrae manually using Wings3D, and then deform these key vertebrae using elastic transformation based on the anatomical features points on each vertebra to fit other vertebrae. In the project, we choose to simplify 3 key vertebrae ($T4, T9, L2$) (Figure 2.14), and fit other vertebrae from these simplified models according to minimum distance principle (See Table 2.3.4).

2.3.5 Pelvis and Sacrum Processing

In the articulated bone structure, the whole spinal column and rib-cage are supported by the sacrum, and the sacrum is supported by the pelvis. We know that the Boston brace which is used to treat the scoliosis needs to interact with the pelvis. Sometimes we should include these parts of bone into the 3D tetrahedral mesh model of human

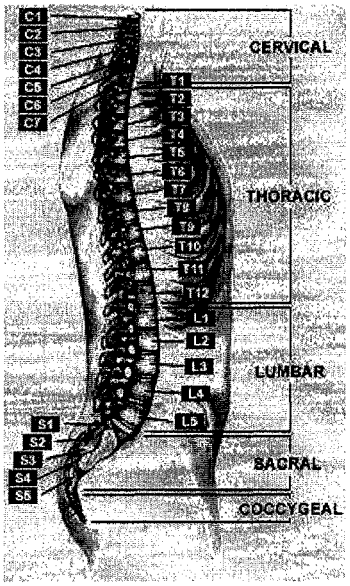


Figure 2.14 Spinal column. (vertebrae $C1 \sim C7$ are not included in our models; vertebra $S1 \sim S5$ are degenerated into the sacrum bone.

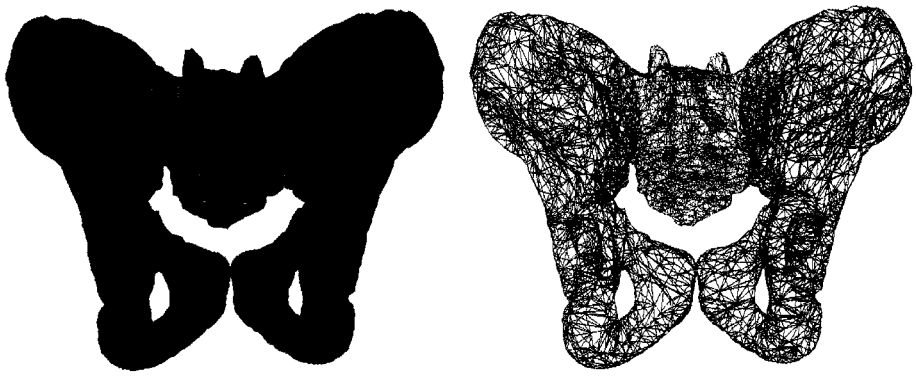
trunk depending on the requirement of Finite Element simulation. Figure 2.15 shows the original 3D pelvis and sacrum model reconstructed by Sainte-Justine Hospital.

In Figure 2.16, we can see the sacrum has complex structure, which have little affection to the Finite Element simulation and make the 3D mesh generation more difficult. We need to simplify these parts of the bone model.

Once again, we use the 3D modeling tools *Wings3D* to manipulate the sacrum model manually to remove the holes in the sacrum surface mesh. Then we use the non-manifold edge removing algorithm and the topology separation algorithm to get the outer surface mesh of the sacrum model. At last, FastRBF was used to produce a watertight regularized mesh.

Although these three bones (left pelvis, sacrum, right pelvis) are physically separated, but in fact they are connected with each other so tightly that we can take them as a single object to simplify the tetrahedron mesh generation procedure.

Of course, merging them into one connected surface mesh is more difficult than separating them since it is related with the geometry instead of topology. First we should



(a) Original model (b) Wireframe view

Figure 2.15 The pelvis and sacrum model reconstructed by Sainte-Justine Hospital. (together 4,509 points)

(a) Original model (b) Transparent view

Figure 2.16 The sacrum have complex structure.

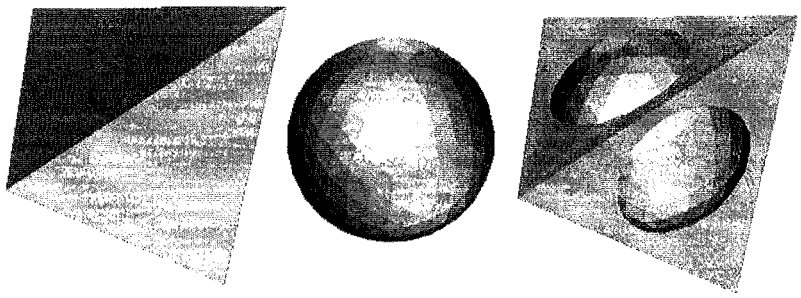


Figure 2.17 Boolean set operation on the surface mesh (Union).

process these three bones to produce the valid watertight surface mesh using FastRBF. And then two methods are available for this merging procedure:

FastRBF: An arbitrary ray which starts from a point inside a polyhedron will have an odd number of intersections with the faces of this polyhedron. Using this property combined with the fast spatial search technique (OCTREE) to determine whether the points of one mesh are located inside another mesh. In practice, degenerated situations may occur, because it is possible that the ray intersect with a vertex, or coincide with a face. In this case, we can initiate more rays in different directions to avoid this problem. Then we delete these conflicting points and their associated edges and triangles. At last we use FastRBF to fill the holes between these three bones to form a single closed mesh.

Boolean Set Operation: We use a robust boolean set operation on the surface mesh to merge these three parts into one single mesh. The open source software package GTS (Appendix III.5) is available to do this work. Figure 2.17 illustrates the idea of boolean set union operation on the meshes of tetrahedron and sphere.

Once the single mesh of the pelvis and sacrum is available, we use the *Wings3D* to simplify the mesh.

2.3.6 Intersection Removal

As we know, the whole human skeletal structure is composed of many separated bones. These parts are physically separated and connected together only by the ligaments. But in fact some bones are so close to each other, that it is very hard to separate them geometrically. In the reconstructed 3D skeletal model provided by Sainte-Justine Hospital (Figure 2.10), there are many undesired intersections between different parts. Before we generate the 3D tetrahedron mesh for this model, all these intersections must be removed manually.

In the previous subsection, we have merged the pelvis and sacrum together into one single mesh since they are combined with each other so tightly. But for the intersection between ribs and vertebrae, we should separate them carefully according to the anatomical structure.

Sometime the rib will break out of the external trunk skin surface. In practice, we delete the intersection part manually or just keep them untouched.

2.4 Multiple Domain Mesh Generation and Ligament Modeling

Actually, there are many different materials inside the human trunk, for example, bone, fat, organ, muscle, ligament, etc. However at present, we just want to consider some main volume regions for simplicity to be sure that we could continue our work without being overwhelmed by an enormous amount of details. The space inside the bone surface is one such volume region (R_{bone}), although it is composed by many disjoint parts. And the space between the external skin surface and bone surface is another volume region (R_{tissue}) (Figure 2.18). As for the ligaments which connect the rib and vertebra, we will choose another method to represent them.

As we have mentioned in subsection 1.4.5, NetGen (Appendix III.4) doesn't support multiple domain mesh generation. Fortunately, we can control the mesh generation procedure step by step within NetGen. Our multiple domain 3D volume mesh generation scheme follows:

Figure 2.18 Two volume regions in the human trunk model: volume region R_{tissue} is between the external trunk skin and internal bone surfaces; volume region R_{bone} is inside the bone surface composed by many disjoint parts. (The external trunk model is partially clipped for the visualization purpose).

1. First, we use NetGen to analyze and process the surface of the internal volume region R_{bone}. We call the optimized surface $S_{internal}$;

2. Second, we generate and optimize the 3D volume mesh for R_{bone}. The result of this step —T_{bone}— is the 3D tetrahedral mesh of the internal volume region R_{bone};

3. Then we analyze and optimize the external trunk skin surface, the result is the optimized surface $S_{external}$, and the corresponding volume R_{trunk}.

4. Reverse the orientation of the triangles in $S_{internal}$ and combine it with $S_{external}$. By this way, we obtain the boundary representation of the volume region between the skin and bone surfaces — R_{tissue};

5. Use NetGen to generate and optimize 3D volume mesh for R_{tissue} while without modifying the boundary faces. We call this 3D tetrahedral mesh T_{tissue};

6. Combine the two 3D tetrahedral mesh T_{bone} and T_{tissue}. We get the whole 3D mesh for the human trunk. This step is a bit difficult, because the two volume

region share the same internal bone surface. We need to find the corresponding points in the two mesh models. Once again, the fast spatial search technique can be used to accelerate this process.

In step 3, we assume the internal bone model are completely inside the external trunk skin surface. However we know that it is not always true. Sometimes the ribs will break out of the external skin surface after registration and fusion. We can handle this problem in two ways:

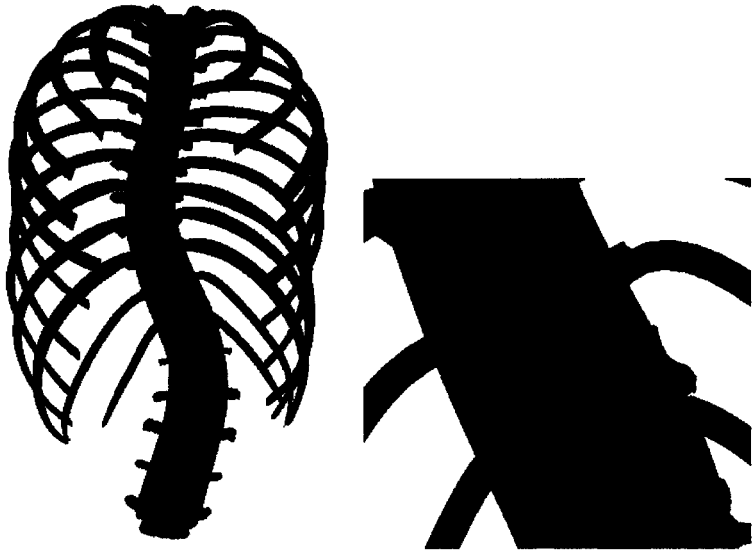
- Removing the outbreaking parts of the bone. This way will facilitate the later 3D mesh generation procedure. However, it needs manual effort and loses some information . We know the rib cage plays a more important role than soft tissue and skin in the Finite Element simulation.
- Keeping these bones unchanged. Then we need to carefully construct the soft tissue region R_{tissue}. we use the boolean set operations in GTS to compute this volume region: $R_{tissue} = R_{trunk} \cap \overline{R_{bone}}$.

As we know, the different parts of bones are connected by ligaments and it is very hard to model ligaments geometrically. So we use a different strategy to construct the ligament region.

A curved tube which represents the ligament region is constructed using Cardinal Spline. We let the center axis of this tube pass through the centroids of each vertebra (computed by the anatomical feature points on each vertebra). And change its radius along the axis path according to the size of the vertebra. For each tetrahedron in the 3D volume mesh T_{tissue}, if it is located inside this ligament tube, then it belongs to the ligament region.By this way, we can effectively and efficiently construct a coarse ligament region which connects the different parts of internal bone model.

2.5 Surgery Prediction

For some severe scoliotic patients, surgeries are needed to get rid of the spinal column curve and prevent deformity. The instruments are often used to correct the position of



(a) Whole view (b) Close view

Figure 2.19 Construct the ligament region—curved tube—by Cardinal spline.

the vertebrae (Figure 2.20). Before the surgery, The surgeons and patient often wish to predict the external trunk surface change.

We propose a simple and fast algorithm to help the surgeon and patient to predict the external trunk shape change after scoliosis surgery. Before the surgery, we can collect the external trunk skin and internal bone surface model from the patient. And we also extract 6 control points for each vertebra according to the anatomical features from the multiple X-ray images during the reconstruction procedure. We intend to apply the Thin-Plate Spline Interpolation (tps_{in}) to simulate the external trunk shape deformation using these control points as source landmark. This algorithm is explained in detail as follows:

1. First, before the surgery we acquire the external trunk skin and internal bone surface models for the patient, and extract the radio-opaque makers position $\{M_{ext}\}$ and $\{M_{in}\}$ in the external and internal model coordinate system space respectively. At the same time we get the control points $\{CP_{pre}\}$ on each vertebra in the internal bone model .

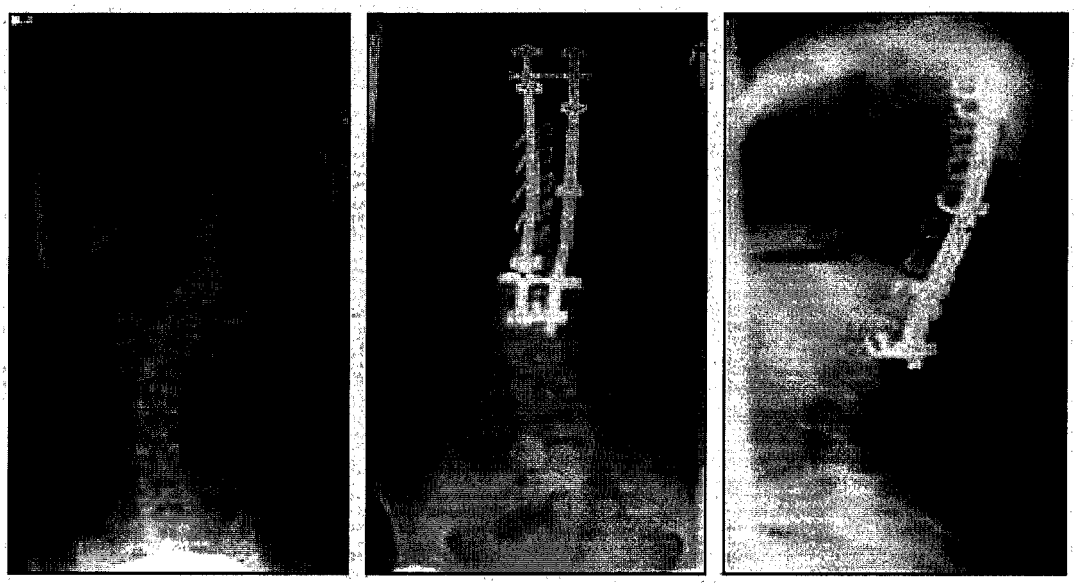


Figure 2.20 Preop and postop segmental spinal instrumentation. Left: back view, before operation; Middle: back view, after operation; Right: sagital view, after operation. (www.espine.com)

2. Then deform the external trunk models by Thin-Plate Spline Interpolation method using $\{M_{ext}\}$ and $\{M_{in}\}$ as source and target landmarks. (In fact, this step is not necessary if we don't want to evaluate the prediction result)

3. The surgeon propose the optimal position of the control points on each vertebra to achieve best treatment effect before the scoliotic surgery. By this way, we get another set of points $\{CP_{post}\}$.

4. Compute the TPS interpolation coefficients with $\{CP_{ori}\}$ and $\{CP_{post}\}$ as the corresponding source and target landmark sets.

5. We deform the transformed external trunk model from step 2 (or original model if we don't want to evaluate the prediction result) using the TPS interpolation with computed coefficients in above step.

To evaluate the prediction result, we intend to compare this predicted external trunk skin model with the real external trunk skin model which is acquired after surgery. We will show our result in the next chapter.

CHAPTER 3

RESULTS AND DISCUSSION

In this chapter, we will show our experimental results and make some discussion. First, we explain our experimental datasets in the section 3.1. Then we show the estimation result of the minimum interval of the optimal trade off parameter for tps_{ap} in subsection 3.2.1. we give the evaluation result of different registration methods in subsection 3.2. The combination result of external trunk skin and internal bone surface models is presented in subsection 3.2.3. In section 3.5 the scoliotic surgery effect prediction result is explained. The preprocessing result of the external and internal surface models is displayed in section 3.3. The multiple domain 3D tetrahedron mesh generation and ligament modeling result are given in the last section.

3.1 Experimental Datasets

From January 2004 to present, we have investigated about 13 patients. however, due to various reasons, there are some little problems in our data set: some patients have only taken 2 external trunk skin models; some have no internal bone models; some external trunk skin models present bad quality texture where we can't recognize the landmarks; and sometimes obvious errors occur on the models. Altogether, 11 patient have 4 useful external models, 1 patient has 3 useful external models; while 3 patients have useful internal models.

For the patient $F0008$, we have collected her external trunk skin and internal bone surface models both before and after surgery. We will use these models to evaluate our surgery prediction algorithm.

3.2 Registration Results

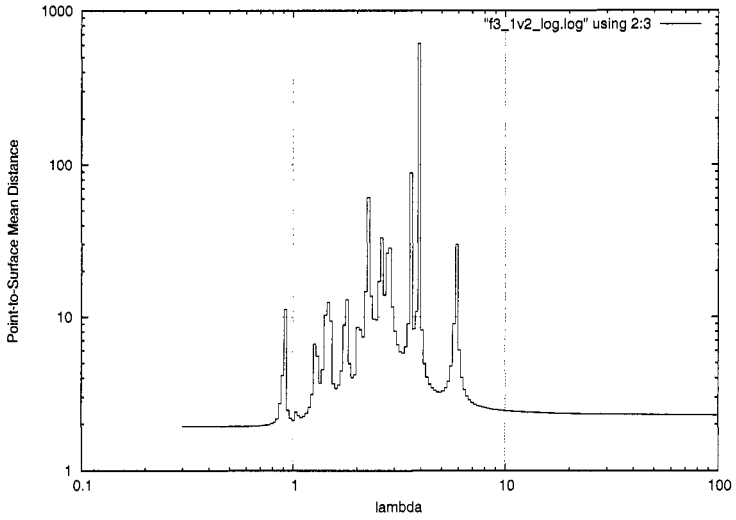
In this section, first we determine the minimum interval for the optimal trade off parameter of the Thin-Plate Spline Approximation. Then we evaluate the transformation results of different registration methods through the comparisons of pairs of external trunk skin models. At last we give the fusion result.

3.2.1 Computation of Optimal Trade off Parameter

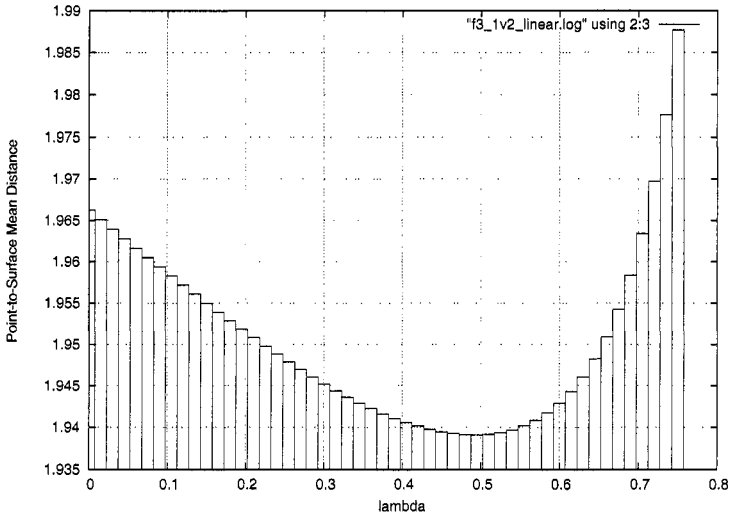
We use the Brent Minimization algorithm to compute the optimal trade off parameter λ for the Thin-Plate Spline Approximation (tps_{ap}) registration scheme. As we have mentioned in subsection 1.2.4 on page 16, the Brent Minimization algorithm needs to know the initial interval which includes the minimum value. We use the brute force method to estimate this interval.

An example of experimental result (Source external model is $F00030000$; Target external model is $F00030001$, and $m = 2$ and $d = 3$, where the meanings of m, d are explained in subsection 1.2.2 on page 13; And the variances of landmark localization errors $\sigma_{i,p}, \sigma_{i,q}$ are $1.1mm$, the external trunk skin surface mesh resolution) shows that $BPTSMD$ varies in a unstable way when λ changes from small value ($\lambda = 0$) to large value ($\lambda = 1000$) (see Figure 3.1(a)). When $\lambda \in [0.8, 8]$, $BPTSMD$ varies dramatically, that is because the tps_{ap} linear system equation (Equation 1.10 on page 16) is ill-conditioned when $\lambda \in [0.8, 8]$, and sometime singularities will happen. For $\lambda > 8$ the curve is smooth, monotonically decreasing, converging to a constant value . When $\lambda < 0.8$, from right to left, the curve first reaches a local minimum value, and then monotonically increases and converges to a constant value. However the right region ($\lambda > 8$) is always higher than the left region ($\lambda < 0.8$). Figure 3.1(b) reveals that when $\lambda \simeq 0.49$ $BPTSMD$ reaches the local minimum value.

From the TPS approximation linear system equation (See Equation 1.10 on page 16), we know that tps_{ap} transformation is stable and $BPTSMD$ will converge to a constant value when λ approaches $\pm\infty$. Based on these facts, we estimate that the optimal trade off value λ should appear in a small region near zero. We have done extensive experiments on the clinical external trunk skin surface models data to estimate this min-



(a) $\lambda \in (0.1, 100)$



(b) $\lambda \in (0, 0.8)$

Figure 3.1 An example of experimental result: $BPTSMD(mm)$ vs lambda (λ); Source model: $F00030000$; Target model: $F00030001$; $m = 2$ and $d = 3$, kernel: r; $\sigma_{i,p} = \sigma_{i,q} = 1.1mm$.(Metric: mm)

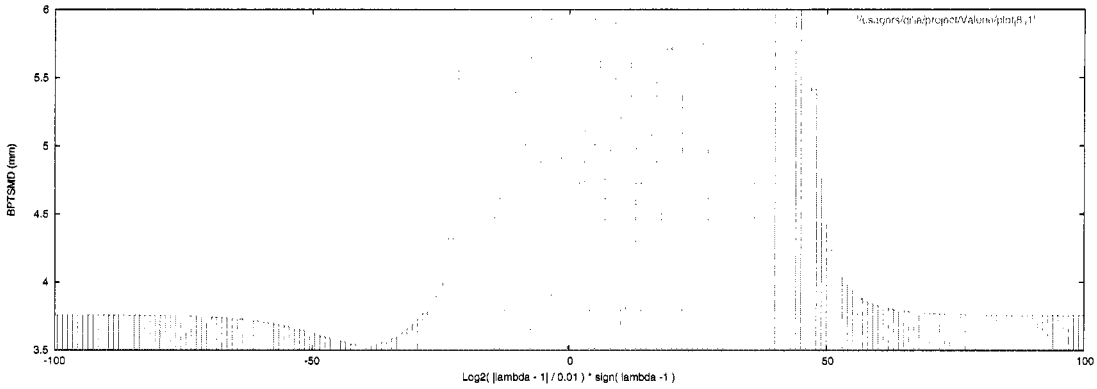


Figure 3.2 Estimate the minimum interval of optimal trade off parameter for tps_{ap} (kernel: r) transformation. Source Model: $F00080003$; Target Model: $F00080001$; $\sigma_p = \sigma_q = 1.1mm$; and y-axis is *BPTSMD* (mm). x-axis is λ with special logarithm scale: $\lambda = 1+sign(i)*c*2^{abs(i)/k}$, where $i = -100, -99, \ldots, 100$, and $c = 0.01, k = 5$.

imum interval. the results show the optimal value actually occurs in the small region $(-4.5, 0.8)$ (See Appendix I.1 for the extensive experiment results). Figure 3.2 illustrates one result of the extensive experiments. Notice that we choose the λ value according to the logarithm equation:

$$\lambda = 1 + sign(i) * c * 2^{abs(i)/k}$$

Where $i = -100, -99, \ldots, 100$, and c, k are the constant variables to control the sampling density. In our project, we let $c = 0.01, k = 5$. Because $BPTSMD$ changes more rapidly in the region near 1 than at other places, more samples are selected from this region.

We input the initial minimum interval $(-15, 1)$ and initial estimated minimum value 0 to Brent Minimization algorithm to fast compute the optimal λ value. In most cases, this algorithm and configuration can return the correct optimal trade off value in about $12 \sim 15$ iterations (the iteration returns when the minimum interval shrinks to 0.01, since in this interval the BPTSMD value change is very small). Figure 3.3 shows an example output of using Brent Minimization algorithm to find this optimal value in 12 iterations. It takes about 120 seconds using the machine in the lab (CPU: 1.8GHZ, MEM: 512M). The optimal λ values for all pairs of external models of the same patients are presented in next subsection.

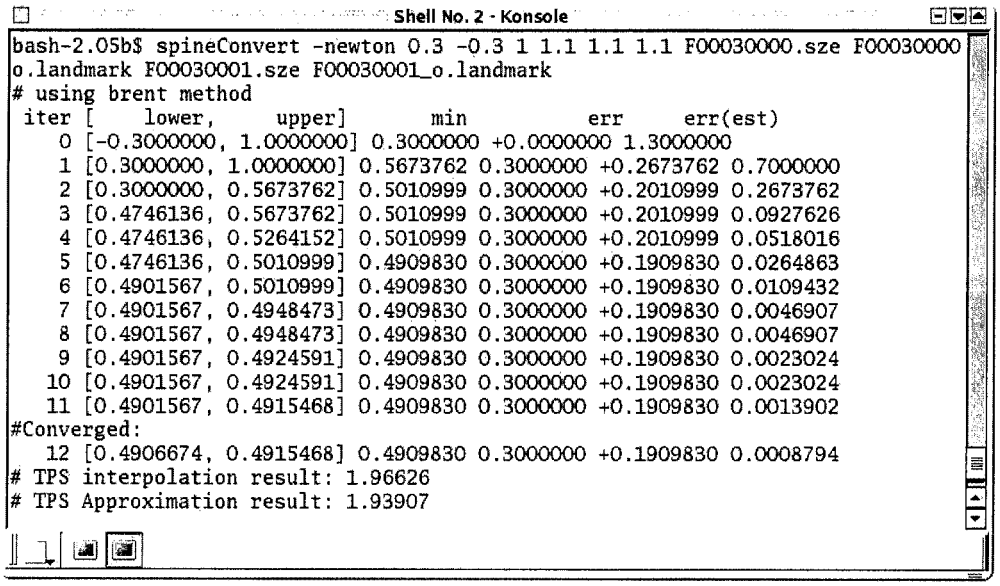


Figure 3.3 An example of experimental result: output of our program using the Brent minimization algorithm to find the optimal trade off parameter in 12 iterations.

3.2.2 Comparison of Different Registration Methods

We evaluate and compare the registration result through two different external trunk skin models from the same patients. As stated before, we have investigated 13 patients. The textures of external trunk skin model of patient $F0013$ are so bad that we can't recognize the markers on the skin. And we have only three external trunk skin models from patient $F0014$. Altogether, we have 47 external models from 12 patients available for this comparison experiment.

Though our main purpose is to deform the external models of posture like Figure 2.1(b) (optimal posture for Inspeck's 3D camera system) while with external models of posture like Figure 2.1(a) (optimal posture for acquiring multiple-view X-ray images) as reference and evaluate the registration result. We still want to do more comparisons on all possible combination pairs of external models to make full use of these valuable clinical datasets.

We register and compare the external models with each other from the same patients. For each patient (4 external models), the comparisons between two external models of

different postures are 8, and comparisons of same postures are 4; The patient F0014 only has 3 external models, we can perform 4 comparisons on different postures, and 2 times on same postures. Altogether we can perform $(12 \times 11) + (6 \times 1) = 138$ times comparisons. For each two external trunk skin models, we register them by $rigid, affine, tps_{in}, tps_{ap}, tps_{in}(r^3)$ schemes ($tps_{in}(r^3)$ is the Thin-Plate Spline Interpolation with kernel r^3 when $m = d = 3$). And also we compute the optimal trade off parameter λ for tps_{ap} transformation. The detailed results are listed in the Table I.2, I.3 and I.4 in the appendix on page 107.

We have gathered some statistical results in the Table 3.1, 3.2 and 3.3. From these tables, we make some observations and discussions as follows:

- First, tps_{ap} has the best registration result, the overall average $BPTSMD$ value is $2.683mm$.
- And tps_{in} also has good performance ($2.717mm$, $\frac{tps_{in}}{tps_{ap}} = 101.3\%$).
- $rigid$ transformation achieves the worst effect without surprise.
- Optimal λ actually appears in the small region $(-4.389, 0.757)$ (see the detailed result table I.2 on page 107). However, it is greatly uncertain (-0.26 $\pm$ 0.929). And there is no obvious relationship between the λ value and the patient. So sometimes we can just use tps_{in} to register the flexible external and internal human trunk models instead of tps_{ap} scheme.

In Table 3.2 and 3.3, we list the same statistical result from the experiments through the external models from the different and same postures, respectively. Except that the average $BPTSMD$ value is smaller in the same posture cases (because the external trunk models from the same posture have little deformation than that from different posture), there are no other obvious difference could be found.

In Figure 3.4 and 3.5, the registration results (error distribution) of different methods are showed.

Note that we assume the all the landmarks have the same localization errors for the TPS approximation ($1.1mm$, equal to the external models resolution). However, this is not

Table 3.1 Statistics of the result. (This table displays the mean $BPTSMD$ value (mm) of the different registration methods for each patients. Second column gives the number of samples considered for this statistical result. Last column shows the mean λ and its standard variances value per patient. The overall statistics results are in the second last row. And we also compute the overall performance ratios of different registration methods compared with tps_{ap} in the last row.)

Patient ID	**Samples**	$rigid$	$affine$	tps_{in}	$tps_{in}(r^3)$	tps_{ap}	λ
F0003	12	2.798	2.581	2.317	3.130	2.285	-0.00 $\pm$ 0.883
F0006	12	3.379	3.259	2.533	3.865	2.524	-0.02 $\pm$ 0.217
G0007	12	2.818	2.664	2.330	2.649	2.310	0.19 $\pm$ 0.353
F0008	12	3.238	3.163	3.001	3.479	2.981	-0.97 $\pm$ 1.149
F0014	6	4.947	3.397	2.863	3.009	2.813	-0.22 $\pm$ 0.610
G0015	12	5.038	4.150	3.456	4.520	3.427	-0.52 $\pm$ 0.602
F0016	12	2.335	2.701	2.471	3.072	2.441	-1.03 $\pm$ 1.578
F0017	12	4.157	3.797	2.584	3.643	2.578	0.01 $\pm$ 0.203
F0025	12	6.255	5.302	2.941	3.028	2.825	0.13 $\pm$ 0.854
F0026	12	3.621	3.361	2.784	3.199	2.766	-0.64 $\pm$ 1.329
F0027	12	3.088	3.440	2.848	3.244	2.802	0.21 $\pm$ 0.542
F0029	12	3.784	2.901	2.553	2.669	2.514	-0.25 $\pm$ 0.592
Total	138	3.738	3.393	2.717	3.305	2.683	-0.26 $\pm$ 0.929
Ratio: $1/tps_{ap}$		139.3%	126.6%	101.3%	123.2%	100%	

true and it could be the reason that optimal trade off parameter λ can't converge to a small range. In the future, we need to do more work to estimate the individual landmark localization errors (for landmarks on both external and internal models).

3.2.3 Fusion of External trunk and Interior Bone

Now we fuse the external trunk model and interior spine model together by different transformation schemes, and evalute the fusion result. Figure 3.6 displays a fusion result for the patient model $F0006$. We can see that in some places the rib bones break out of the trunk skin surface. In figure 3.7, we show the parts of bones which break out of the external trunk skin. And we compute the volume of these outbreaking parts. In Table 3.4, we list the results to compare different registration methods. However we just have only 3 patients available (with internal bone surface models) for this experiment and no rule can be found from these limited amount of results. At present, we can't tell whether there exists a link between the volume of the outbreaking rib bone and the registration method.

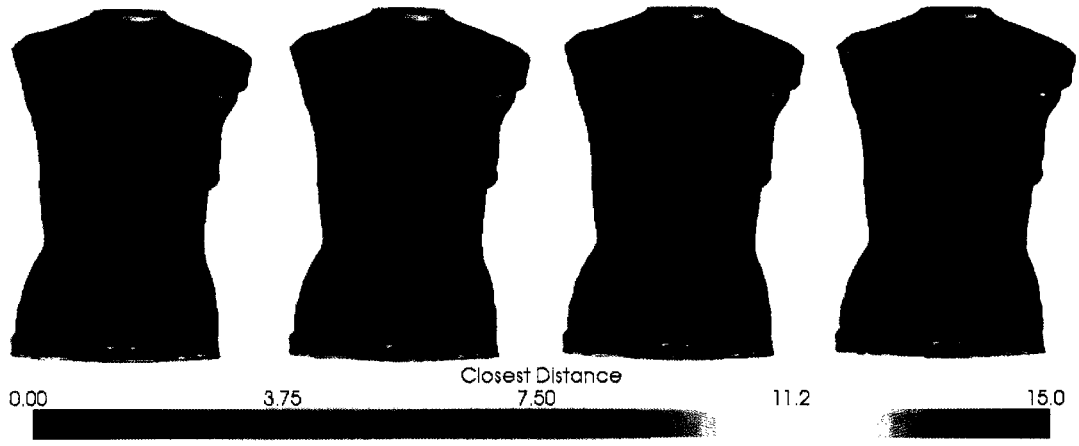
In the Figure 3.8, a part of pelvis bone is outside of the external surface. This result was

Table 3.2 (Continued, See Table 3.2.2 for explanation). The source and target external models are from different postures.

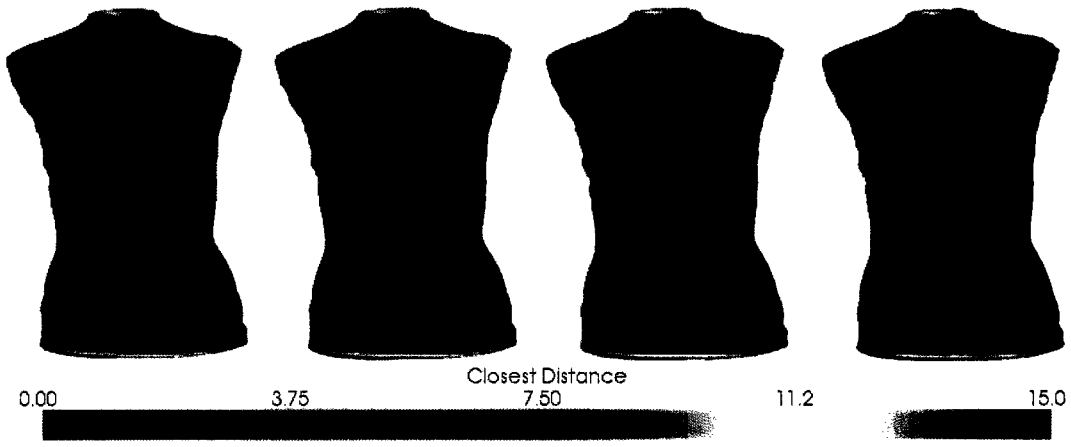
Patient ID	**Samples**	$rigid$	$affine$	tps_{in}	$tps_{in}(r^3)$	tps_{ap}	λ
F0003	8	3.423	3.131	2.864	3.594	2.829	-0.31 $\pm$ 0.951
F0006	8	3.666	3.670	3.016	4.442	3.010	-0.07 $\pm$ 0.234
G0007	8	3.153	3.038	2.651	3.064	2.635	0.10 $\pm$ 0.390
F0008	8	4.048	3.935	3.718	4.289	3.699	-0.48 $\pm$ 0.991
F0014	4	6.213	4.283	3.626	3.281	3.574	0.12 $\pm$ 0.393
G0015	8	5.815	4.859	4.097	5.402	4.053	-0.80 $\pm$ 0.548
F0016	8	2.748	3.154	3.041	3.799	3.004	-1.66 $\pm$ 1.601
F0017	8	4.638	4.264	2.808	3.921	2.804	0.06 $\pm$ 0.123
F0025	8	8.305	7.003	3.733	3.649	3.572	0.57 $\pm$ 0.153
F0026	8	4.071	3.916	3.183	3.652	3.174	0.04 $\pm$ 0.198
F0027	8	3.755	4.316	3.505	4.037	3.439	0.44 $\pm$ 0.075
F0029	8	4.258	3.584	3.118	3.081	3.071	0.14 $\pm$ 0.196
Total	92	4.434	4.088	3.265	3.876	3.224	-0.17 $\pm$ 0.863
Ratio: $1/tps_{ap}$		135.5%	126.8%	101.3%	120.2%	100%	

Table 3.3 (Continued, See Table 3.2.2 for explanation). The source and target external models are from same posture.

Patient ID	**Samples**	$rigid$	$affine$	tps_{in}	$tps_{in}(r^3)$	tps_{ap}	λ
F0003	4	1.546	1.483	1.222	2.202	1.198	0.61 $\pm$ 0.077
F0006	4	2.805	2.438	1.565	2.710	1.552	0.09 $\pm$ 0.144
G0007	4	2.146	1.915	1.686	1.820	1.658	0.35 $\pm$ 0.212
F0008	4	1.620	1.618	1.566	1.858	1.544	-1.94 $\pm$ 0.814
F0014	2	2.415	1.626	1.338	2.466	1.293	-0.90 $\pm$ 0.009
G0015	4	3.486	2.732	2.175	2.757	2.175	0.04 $\pm$ 0.045
F0016	4	1.509	1.795	1.331	1.618	1.316	0.22 $\pm$ 0.035
F0017	4	3.194	2.862	2.136	3.087	2.126	-0.08 $\pm$ 0.313
F0025	4	2.156	1.900	1.358	1.785	1.332	-0.74 $\pm$ 1.043
F0026	4	2.721	2.251	1.987	2.293	1.950	-1.99 $\pm$ 1.648
F0027	4	1.755	1.687	1.535	1.658	1.527	-0.23 $\pm$ 0.820
F0029	4	2.836	1.534	1.423	1.846	1.400	-1.02 $\pm$ 0.098
Total	46	2.346	2.002	1.622	2.162	1.602	-0.45 $\pm$ 1.034
Ratio: $1/tps_{ap}$		146.2%	125.0%	101.2%	135.0%	100%	

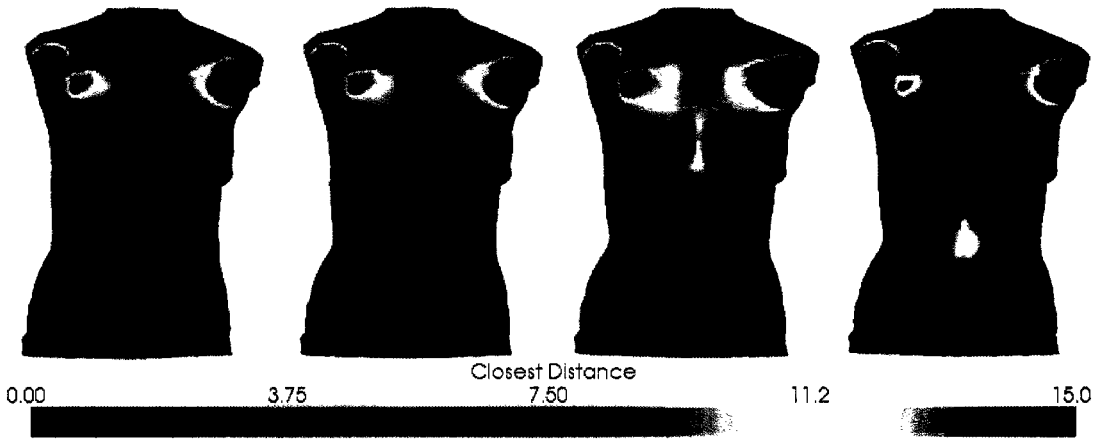


(a) Front view (from left to right: $tps_{ap}, tps_{in}, affine, rigid$).

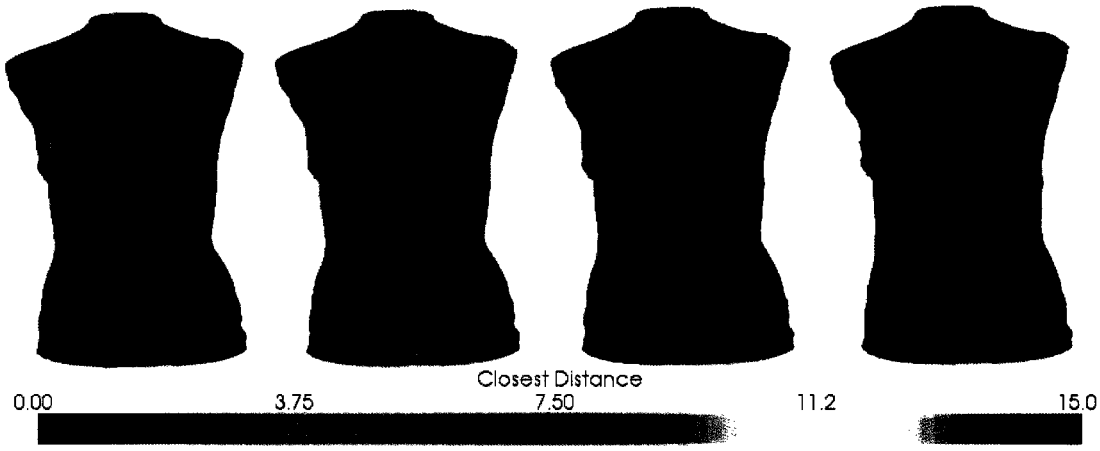


(b) Back view (from left to right: $rigid, affine, tps_{in}, tps_{ap}$).

Figure 3.4 Comparison of different registration methods through the external trunk skin models from the same posture. (Source Model: $F00030000$; Target Model: $F00030001$; Metric: mm).



(a) Front view (from left to right: $tps_{ap}, tps_{in}, affine, rigid$).



(b) Back view (from left to right: $rigid, affine, tps_{in}, tps_{ap}$).

Figure 3.5 Comparison of different registration methods through the external trunk skin models from the different postures. (Source Model: $F00030000$; Target Model: $F00030002$; Metric: mm).

Table 3.4 The comparison of the volume of rib bone which breaks out of the skin surface by different transform methods. ($\lambda = 0.3$ for tps_{ap}).

(mm^3)	posture	$rigid$	$affine$	tps_{in}	tps_{iap}
$F0006$	0	3178	505	405	422
	1	967	618	830	914
	2	4626	1194	451	399
	3	3466	1900	1222	1196
$F0008$	0	8963	7578	13824	14748
	1	13696	11801	16613	17499
	2	16470	5359	6223	6756
	3	13980	5282	7092	7789
$F0014$	0	0	0	0	0
	1	0	0	0	0

obtained using the tps_{in} from model $F00140000$.

There are several possible reasons for these annoying results:

1. When we take the external model by Inspeck 3D cameras and internal model by Multiple X-ray images, we don't ask the patient to hold breath and no repositioning apparatus is used. In most of the fusion results, we can notice that the ribs always break out of skin surface in the region in the front of trunk, where the trunk shape is most easily affected by breathing.

2. The original specimen model which our interior model was deformed from is not precise enough. Figure 3.9 shows an example where one landmark is located inside the pelvis bone. In fact almost all the models have the same problem in the pevlis region.

3. The number and distribution of the landmarks affect the registration result. In our project, we put less radio-opaque markers in the front region of the human trunk than the back region due to identification difficulty.

4. Although the elastic transformations (tps_{in}, tps_{ap}) are very good, they still can't simulate the human trunk deformation completely.

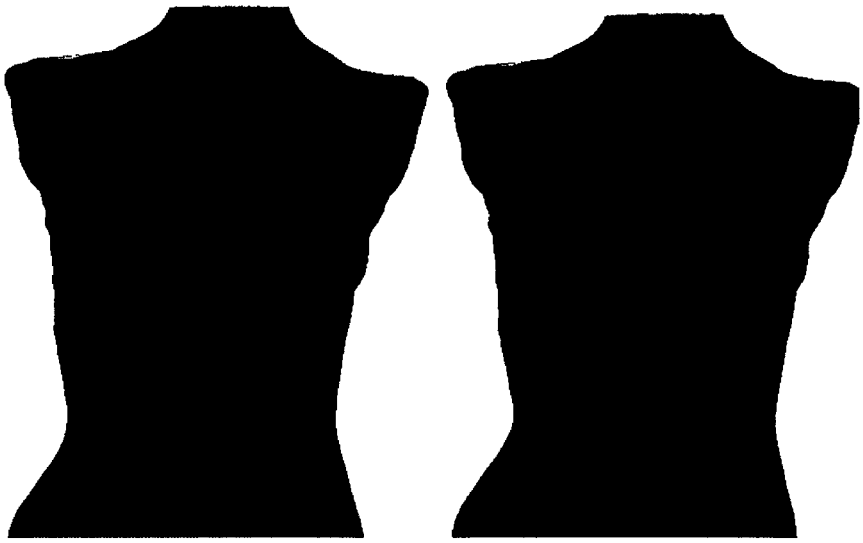
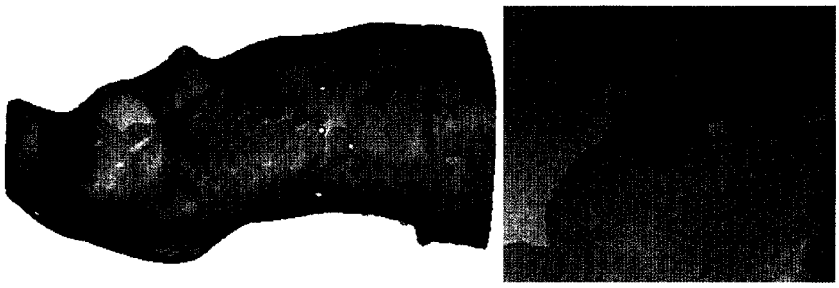


Figure 3.6 Combination result: ribs break out of the skin surface (from external model $F00060003$). Left: Thin-Plate Spline Interpolation transform result; Right: rigid transform result;

Figure 3.7 Combination result: the rib bone which break out of the skin surface by rigid method. Volume: $3178mm^3$. (from external model $F00060000$)



(a) Whole view (b) Close view

Figure 3.8 Combination result: pelvis break out of the skin surface. (from external model $F00140000$)

(a) Part of the pelvis

(b) Close view of the right down corner landmark

(c) Another close view of different direction

Figure 3.9 The obvious bad modelling of the pelvis. This landmark's position is always inside of the pelvis bone for most models.

3.3 Preprocessing Results

In this section, we list a lot of figures to show our results. First we give the processing results of the external trunk skin model by FastRBF. Then we present the processing result of the internal bone surface.

3.3.1 Results on the External Model

First we use the Non-manifold edge removing algorithm to detect the non-manifold edges (Figure 3.10) and delete all the non-manifold edges and their associated triangles from the original model (Figure 3.11). Then we input this modified external trunk skin model into FastRBF to produce a closed (Figure 3.12 and Figure 3.13), smooth (Figure 3.14 and Figure 3.15), simplified (3.16) and regularized (Figure 3.17) boundary representation (B-rep) of the external trunk volume.

In Figure 3.15, Figure 3.16, Figure 3.17, we also compare the FastRBF with the Triangle Decimate algorithm (Schroeder92). According to the survey (Luebke01), Triangle Decimation excels at eliminating redundant geometry, and is suitable for complex organic forms. From these figures we can easily see the results of FastRBF are better than Triangle Decimate algorithm.

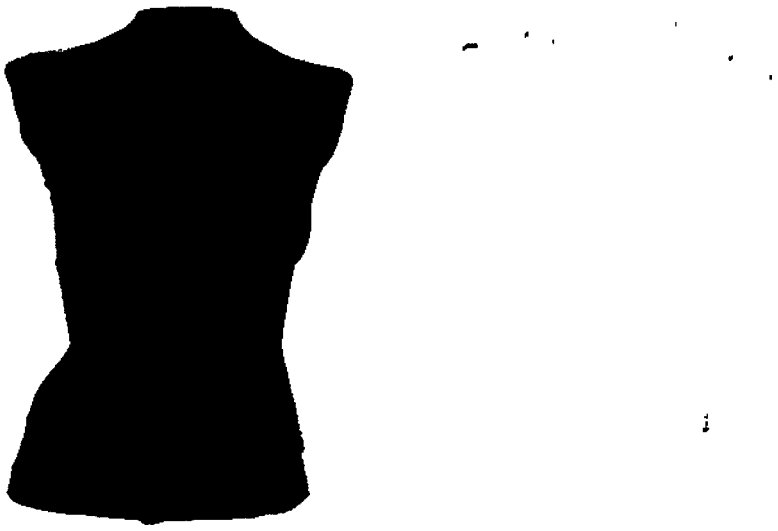


Figure 3.10 Extract the non-manifold edge from the original trunk skin surface model (F00060000). Left: original external trunk surface model; Right: non-manifold edges.

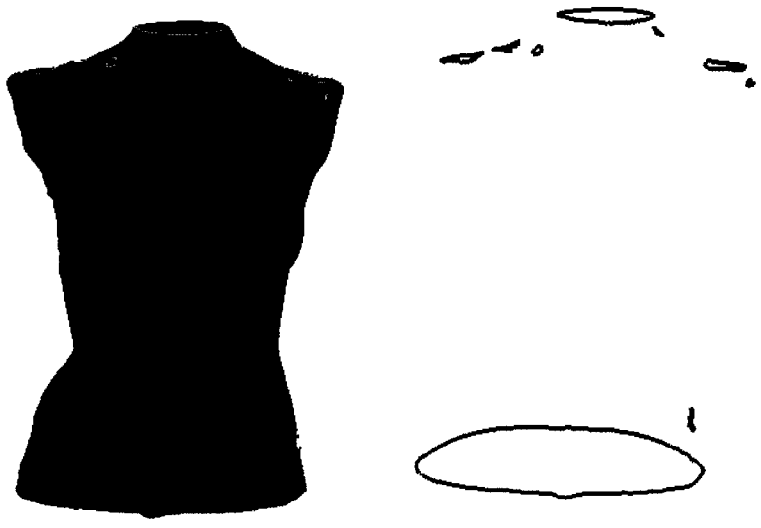


Figure 3.11 New external trunk surface model with more and bigger holes after removing all the non-manifold edges and its associated triangles (F00060000). Left: new trunk mode; Right: more and bigger holes on this new mesh.

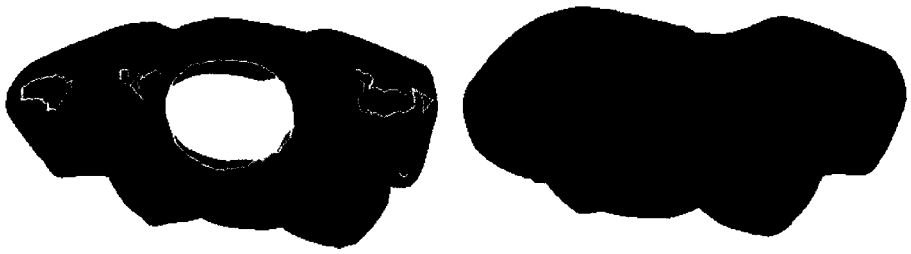
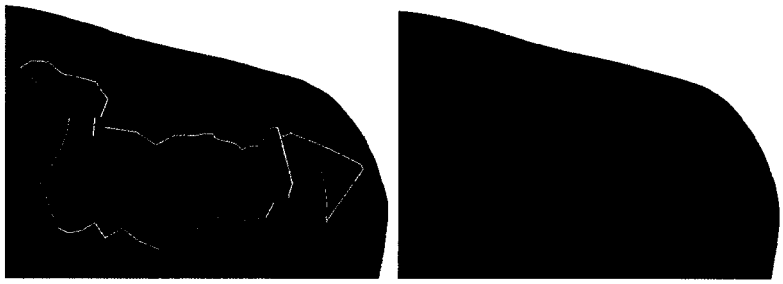


Figure 3.12 Holes filling result by FastRBF (F00060000). Left figure: original model (72610 points); Right figure: result watertight model by FastRBF (51883 points).



(a) Hole on mesh (b) Result watertight mesh

Figure 3.13 Close view of the hole filling result by FastRBF (F00060000). Left shoulder part of the trunk model in Figure 3.12.

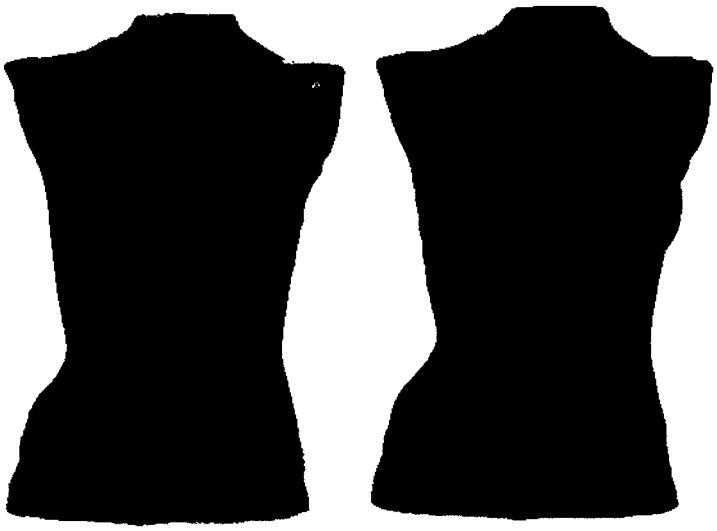


Figure 3.14 Result smooth mesh by FastRBF ($F00060000$). Left figure: original model (72610 points); Right figure: result smooth model by FastRBF (51883 points) .

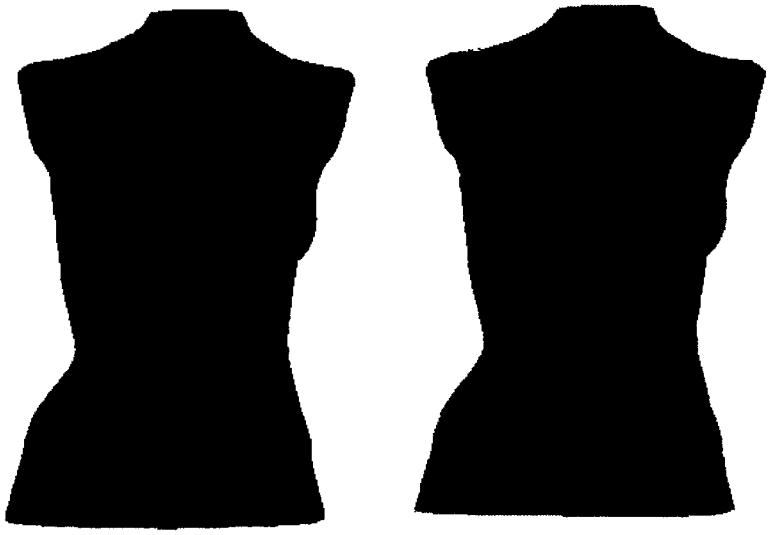


Figure 3.15 Comparison 1: Mesh smoothness ($F00060000$). Left figure: result by FastRBF (5219 points, $SR = 92.8\%$); Right figure: result by Triangle Decimate (5156 points, $SR = 92.9\%$). SR: Simplification Ratio.

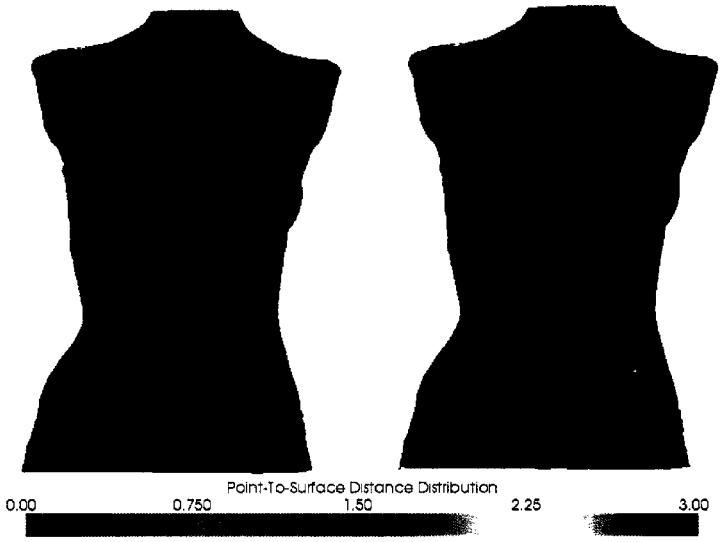


Figure 3.16 Comparison 2: Distribution of the error distance compared with original model ($F00060000$). Left figure: result by FastRBF (5219 points, $SR = 92.8\%$); Right figure: result by Triangle Decimate (5156 points, $SR = 92.9\%$) .

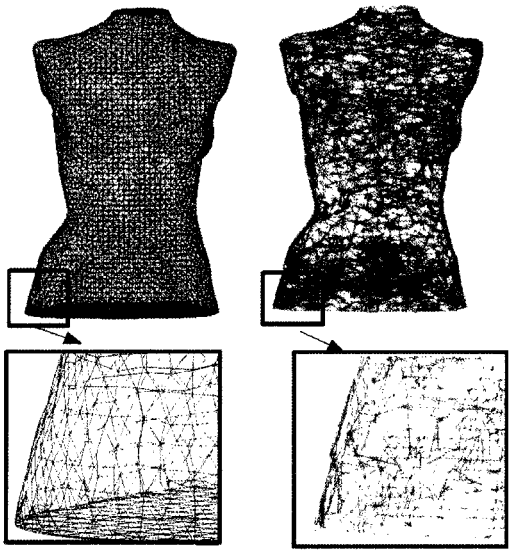
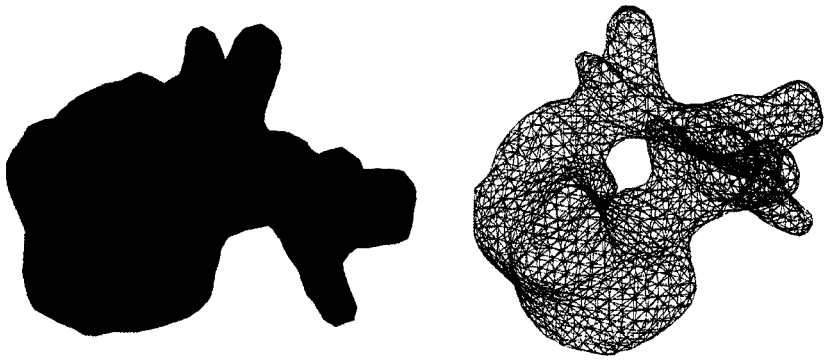


Figure 3.17 Comparison 3: Regularity of the vertexes placement (from external model $F00060000$). Left figure: result by FastRBF (5219 points, $SR = 92.8\%$); Right figure: result by Triangle Decimate (5156 points, $SR = 92.9\%$) .



(a) Watertight and regularized vertebra model by FastRBF

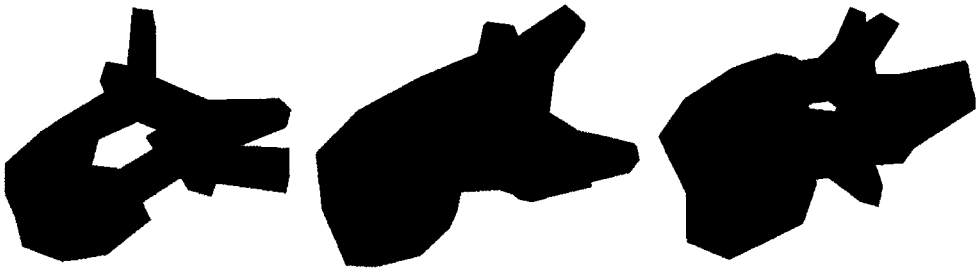
(b) Wireframe

Figure 3.18 Valid watertight and regularized vertebra surface mesh model from FastRBF.

3.3.2 Results on the Internal Model

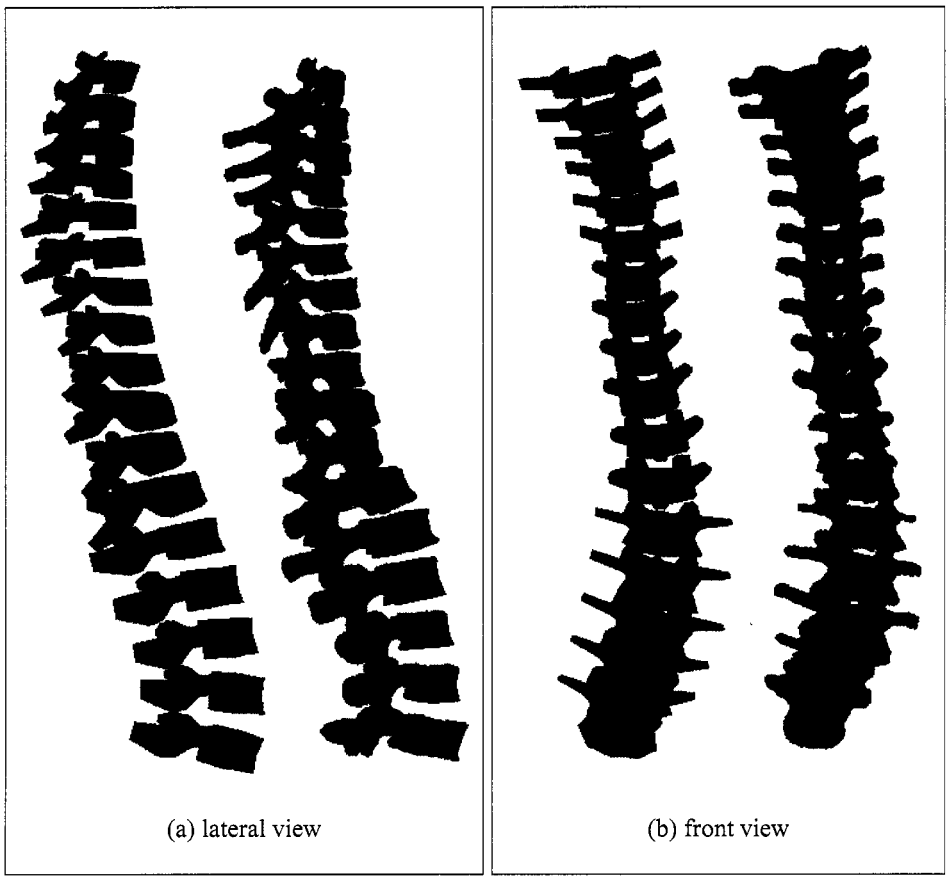
Once again, we remove the non-manifold edges and fill holes on the vertebrae models (Figure 3.18) . Then we use Wings3D to simplify the vertebrae T4, T9 and L3 manually (Figure 3.19). We use these three vertebra as the base models, and deform them to fit other vertebra. See the Table 2.4 for the corresponding deformation relationship. Figure 3.20 displays the processing results of the whole spine. Compared with the original model, we have reduced the number of points by 90.8% while keeping the original shape with good accuracy. In future work, all the vertebrae could be simplified directly without this deformation procedure in order to improve the accuracy as much as possible. In Figure 3.21, the simplified sacrum and pelvis are displayed. Notice that they are merged into one single surface mesh model.

We have to admit that all these works are very very complex and tedious, lots of manual efforts are required to get the final results. Anyhow the original internal bone models must be improved. Once we get a valid, precise and simplified model for one patient, we can easily deform it based on the landmarks transformation to fit other patients. Actually the colleagues in our lab have began to construct a new human trunk model from MRI images (from Visual human project).



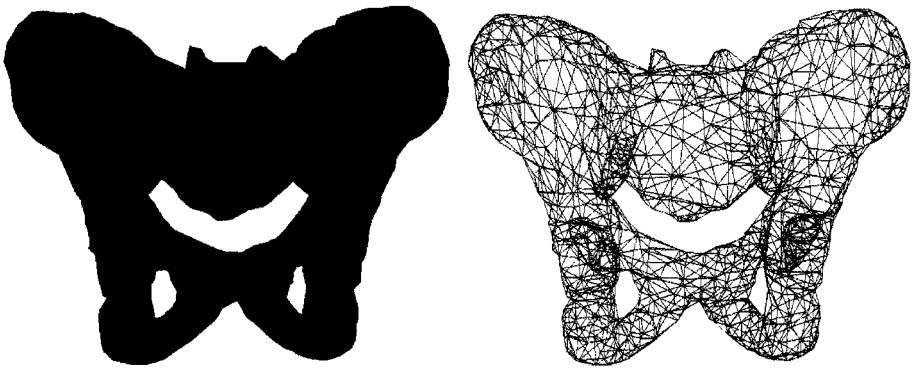
(a) T4 (93 points) (b) T9 (124 points) (c) L2 (108 points)

Figure 3.19 Simplified vertebra model using Wings3D manually.



(a) lateral view (b) front view

Figure 3.20 Simplified watertight spinal column surface model (left figure, 1,476 points, $SR = 90.8\%$) versus original model (right figure, 16,037 points).

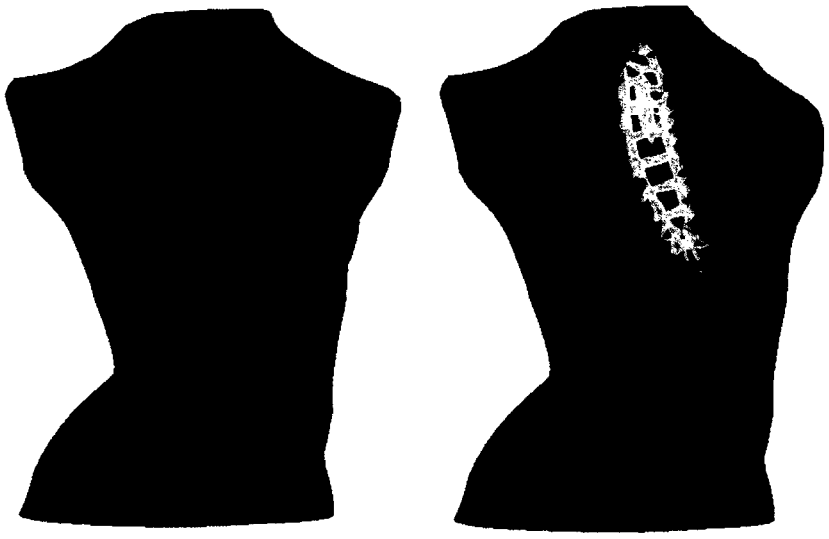


(a) New model of pelvis and sacrum (b) Wireframe view

Figure 3.21 The single watertight, simplified mesh model for the pelvis and sacrum (1,016 points, $SR = 77.5\%$ compared with orignal 4,509 points).

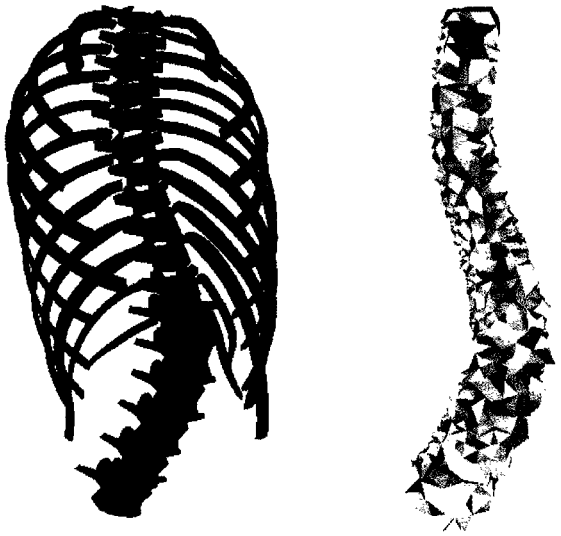
3.4 3D Mesh Results

After registration, preprocessing, fusion, now we have obtained the valid boundary representation of the multiple volume region of the human trunk. We use the multiple domain mesh generation algorithm based on NetGen to generate the precise personalized 3D tetrahedral mesh of the human trunk. See Figure 3.22. Note that in this model, we keep the outbreaking rib bone unchanged. Table 3.5 and 3.6 list the quality statistical result of this 3D mesh . Since our multiple domain mesh generation algorithm needs to keep the internal bone surface (which is shared by two volume regions) without modification during the 3D volume mesh generation procedure, this limits the volume mesh generation and optimization ability. We can see from table 3.6 that there are a few tetrahedrons which have bad shape (in terms of radius ratio of circumsphere and insphere). Fortunately, the extremely bad tetrahedrons often appear at the intersecting places of rib with skin. We can just remove them without affecting the Finite Element simulation. This 3D mesh of human trunk has been tested with Finite Element Simulation successfully in the context of surgery prediction by the colleague in our lab.



(a) Whole Model

(b) Partial view via cut plane



(c) spine and rib-cage. (Tetrahedron: 20689; Point: 9464)

(d) ligament region. (Tetrahedron: 16737; Point: 4878)

Figure 3.22 The personalized 3D tetrahedral mesh of the human trunk (from external model $F00080003$). (including spinal column, rib cage, ligament and soft tissue; Tetrahedrons: 124141; Points: 22334).

Table 3.5 The tetrahedron volume statistics in the mesh ($F00080003$) (Tetrahedrons: 124141).

range (mm^3)	number of tetrahedron	ratio
$0.0001 \sim 0.001$	314	0.25%
$0.001 \sim 0.01$	1392	1.12%
$0.01 \sim 0.1$	2502	2.01%
$0.1 \sim 1$	3592	2.89%
$1 \sim 10$	20079	16.17%
$10 \sim 100$	66334	53.43%
$100 \sim 1000$	28428	22.90%
$1000 \sim 10000$	1495	1.20%

Table 3.6 The distribution of the radius ratio $\frac{R_{circumsphere}}{R_{insphere}}$ of the tetrahedrons in the mesh ($F00080003$) (Tetrahedron: 124141).

range	number of tetrahedron	ratio
≤ 1	1458	1.17%
$1 \sim 10$	12161	9.79%
$10 \sim 100$	109333	88.07%
$100 \sim 1000$	1031	0.83%
$1000 \sim 10000$	88	0.07%
$10000 \sim 100000$	42	0.03%
$100000 \sim 1000000$	16	0.01%
≥ 1000000	12	0.009%

3.5 External Trunk Shape Prediction Result

To evaluate the external trunk shape prediction result, we have collected the external and internal models both before and after scoliotic surgery from patient $F0008$.

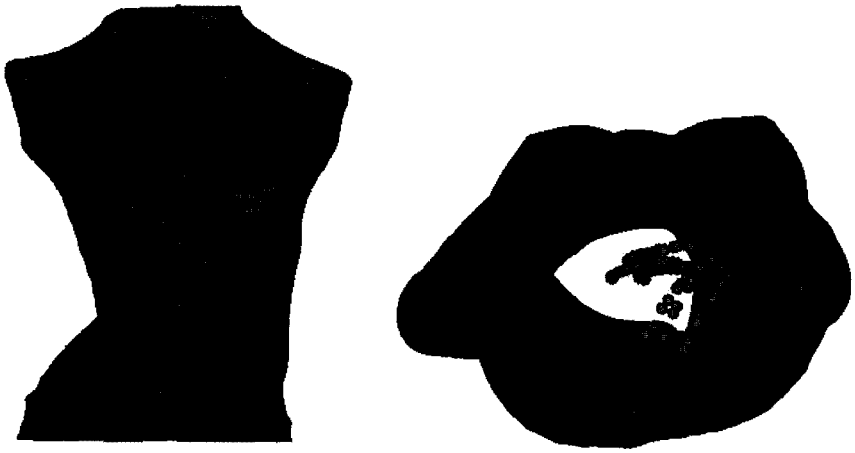
According to the surgery prediction algorithm, first we register the external and internal models acquired before surgery and get the transformed external trunk model — T_{before}. We do the same thing for the models acquired after surgery for comparison purposes, the transformed external trunk model — T_{after}. See Figure 3.24.

Then we use the anatomical feature points on each vertebra acquired before and after surgery as source and target landmarks, respectively, to perform the Thin-Plate Spline Interpolation on the model T_{before}, and we get the prediction result — $T_{prediction}$.

At last, we compare the $T_{prediction}$ with T_{after} to evaluate the surgery prediction result. The distribution of the point-to-surface minimum distance is illustrated in Figure 3.25. The **Bidirectional Point-to-Surface Mean Distance** is $8.5mm$.

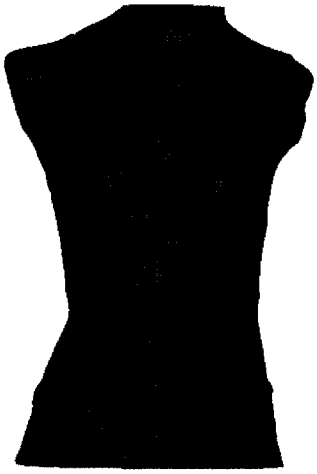
The prediction result is plausible. And the whole procedure is very simple and fast. This algorithm could be used to help the surgeon in surgery plan. In the future, we will do more experiment to verify the effectiveness of this prediction through more clinical data. And after we find the optimal trade off parameter for the human trunk, we can also the TPS approximation to consider the landmark localization error.

However, due to the imitated information (102 pairs of landmarks: 6 points $\times$ 17 vertebrae) which we can use for the Thin-Plate Spline Interpolation, we can't improve the prediction precision any more. That is why we still need to use the Finite Element method to simulate the spinal deformity, which in theory promises higher precision .

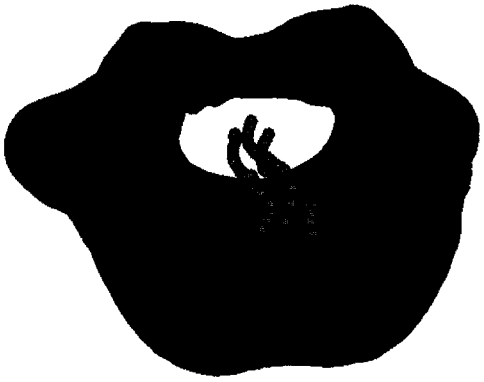


(a) front-back view (pre)

(b) down-up view (pre)

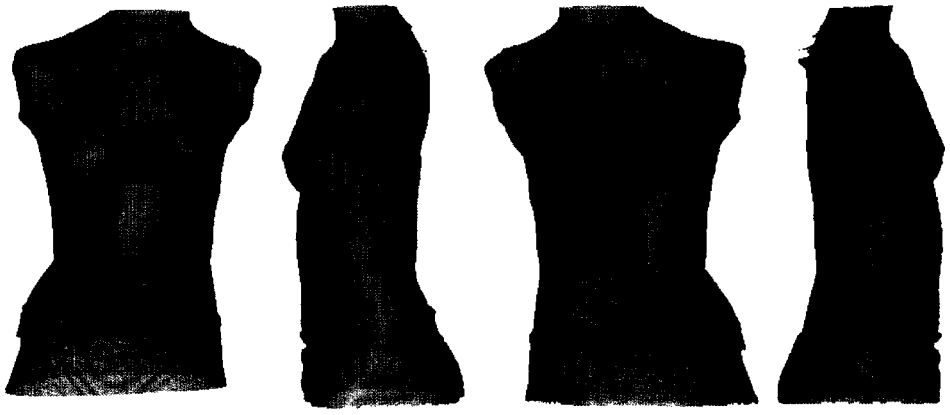


(c) front-back view (post)



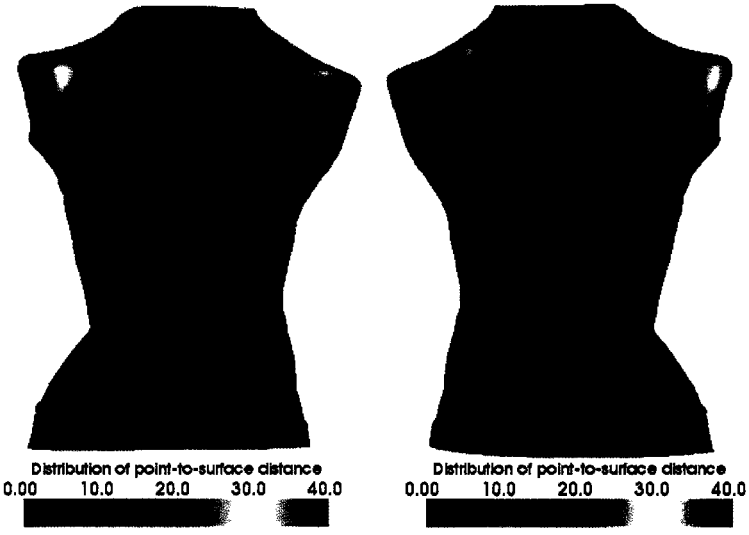
(d) down-up view (post)

Figure 3.23 The external trunk model after registration based on the radio-opaque landmarks on the skin. Magenta balls in right figures present the anatomical feature points on each vertebra acquired from multiple X-ray images. pre: before surgery, $F00080003$; post: after surgery, $F20080003$.



(a) front (b) left side (c) back (d) right side

Figure 3.24 The superimposed external models: gray and transparent model is taken after surgery ($F2080003$); magenta model is the prediction result from $F00080003$.



(a) front view (b) back view

Figure 3.25 Evaluation of the surgery prediction effect ($F00080003$). Mean error Distance: $8.5mm$.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 Conclusion

The final objective of this research is to construct a precise personalized 3D tetrahedral biomechanical mesh of the human trunk from the clinical datasets for Finite Element Simulation of scoliosis treatment. This task has been proved to be extremely complex due to the complicated anatomical features of the human trunk and the limitation of current technologies. Since we prefer to use the non-invasive and affordable techniques to acquire the anatomical structures of the human trunk, we use the Inspeck's 3D camera system to capture the external trunk skin surface models, and the internal bone surface model is reconstructed from multiple-view X-ray images.

The first problem we need to solve is how to fuse the external and internal models of the human trunk since they are acquired at different times, in different sites, by different methods and even with different postures. Another problem we have encountered is that the external skin and internal bone surface mesh models have poor quality in the sense that they don't form a valid Boundary representation of volume, which is required by most 3D tetrahedron mesh generation algorithms.

In this project, we focused mainly on solving the first problem: align and fuse the flexible external and internal models using a landmark-based registration method. We propose to use the local elastic deformation (Thin-Plate Spline Interpolation / Approximation) to register the external and internal models into the same coordinate system. Although these methods have been used widely in the medical image fields, no report was found that they have been applied to register the external trunk skin and internal bone surface models as in our application.

The registration errors come from two factors: the difference of sensors and difference of postures. And the later factor introduces more effect than the first. Since it is hard to directly evaluate the registration errors through the external and internal models that are

different objects, we proposed an approach to assess the deformation due to the difference of postures by transforming and comparing pairs of external trunk surface models from different postures. Especially, we have designed and implemented an evaluation metric — **Bidirectional Point-to-Surface Mean Distance** (**BPTSMD**) — to evaluate and compare the registration results. This metric is plausible due to the fact that the vertices on the external skin surface mesh models have dense and uniform distribution. And it is also very convenient to observe the distribution and contour of the registration errors over external trunk surface. Our experiments results on the clinical datasets from 12 patients show that the elastic methods (TPS interpolation) are much better than the global transformations (rigid, affine). And we have also investigated the TPS approximation schema to account for the landmark localization errors. The results actually show some improvement over the TPS interpolation method. The optimal trade off parameter for TPS approximation has been investigated and located in the context of the human trunk using the Brent Minimization algorithm in an interval that we find through extensive experiments. However, we can't find an obvious pattern of the optimal trade-off parameter distribution. Since currently we just assume that all the landmarks have the same localization errors, and this could be one of the reasons why the optimal parameter can't converge.

Besides the above registration and fusion works, we also have done some method investigation and development work to solve the second problem that we have encountered in this project. We have investigated the existing hole filling algorithms for the surface mesh. A series of algorithms and techniques have been used to produce a valid, precise, watertight, smooth and simplified boundary representation of the external human trunk model. Especially we have chosen the FastRBF as our primary tool to handle the external human trunk surface mesh model. Currently the internal bone surface model can't satisfy the requirement of the 3D mesh generation algorithms. We have spent a lot of time to produce a coarse but valid internal bone surface model, since we need to go through it to construct a coarse 3D tetrahedron mesh model of the human trunk, which is required by the colleague in our lab for the initial Finite Element Simulation of the spinal deformity. |A lots of tedious work have been done manually to get this model. And many drawbacks and suggestions have been reported to improve the internal bone model.

Finally we have proposed a multiple domain 3D volume mesh generation approach based on NetGen (an open source 3D mesh generation software), to generate the 3D tetrahedral mesh of human trunk. However, this personalized 3D tetrahedral mesh model of the human trunk is still a very coarse model. The detailed internal anatomical information of the soft tissue, muscle and ligament inside the human trunk are simplified or just ignored. More work is needed to improve this model.

4.2 Future Work

There are several possible ways and directions to improve our work in the future.

First, more work is needed to investigate the relationship between the distribution of the landmarks and registration effect. We know that more landmarks will improve the registration precision in theory. However we can't increase the number of the markers too much, since it will be hard to establish the corresponding relationship of the markers positions in the multiple-view X-ray images when there are too many markers, and sometimes the marker will be blocked from X-ray by the bone structures and can't be identified. On the other hand, we know that the back shape of the human trunk reveals more information of the spinal deformity than the front part since it is nearer to the spine column, while the shape of the front part is more flexible and easily being changed due to the breathing and postures than the back part. If we put more markers on the back part, then we can preserve more spinal deformity information. If we put more markers on the front part, then we will probably be able to improve the overall registration precision. Because we can't increase the total number of markers too much, here we can see that there is a trade off between preserving more of the deformity information and improving the overall registration precision in choosing the optimal distribution of the markers on the back and front parts of the human trunk. In this project, the marker placements are chosen according to some anatomical features and experiences. In the future, we need to investigate the relationship between the markers placement and the registration effect.

Currently we assume each landmark has the same localization errors (1.1mm, equal to the external surface mesh resolution). However, it is not true and this assumption more or less affected our registration result of the TPS approximation scheme and the

determination of the optimal trade off parameter. In order to better determine the optimal trade off parameter, more work needs to be done to estimate each landmark's localization error in the future.

For computing the **Bidirectional Point-to-Surface Mean Distance** metric more reliably, we construct the region of interest by using the planes that pass through the lowest and highest landmarks and parallel with the XZ-plane. By this way we don't clip away the regions near the arms and shoulders where the manual errors are larger than other place (see Figure 2.7), since the visibility of these parts from the Inspeck's 3D optical camera system is limited. We should devise a more plausible clipping scheme to remove these regions to improve the reliability of our evaluation metric.

In this project we didn't directly evaluate the registration and fusion error of the external and internal surface models of the human trunk. We just estimated the registraion errors from the difference of postures through transforming pairs of external surface models from different posture A and B (see Figure 2.1). However, we know that the external surface models from the posture A are not good (note that the posture A is the optimal posture for acquiring the internal bone model, while it is not good for acquiring the external surface model due to the limited visibility of the optical cameras). That means we have to introduce some extra evaluation errors though we have known it. In the future, we hope to get some high quality external and internal models of the human trunk that are taken at the same time by other techniques (for example, MRI) to assess the elastic registration. Several invariant parameters will be extracted from the entire 3D model of the trunk obtained from MRI to characterize the relative position of the bone structures with respect to the external surface of the trunk. These parameters will be computed from the fused external skin and internal bone surface models of the human trunk acquired via Inspeck's 3D camera system and multiple-view X-rays images respectively. The invariance of these parameters to the posture of the patient is very important because the MRI images are acquired in a supine position.

In fact, the idea of introducing some high quality "standard" models of the human trunk is not limited to the elastic registration purpose, but could also be useful for inferring the internal anatomical structures (organ, muscle, bone etc) of the human trunk, which is very important for our final objective — constructing precise personalized biomechanical 3D tetrahedron mesh of human trunk.

REFERENCES

AMENTA N, BERN M, ET AL. (1998). *A new Voronoi-Based Surface Reconstruction Algorithm.* Proc. SIGGRAPH'98, ACM, 1998.

ANDRE B, DANSEREAU J, LABELLE H. (1994). *Optimized Vertical Stereo Base Radiographic Setup for the Clinical Three-Dimensional Reconstruction of the Human Spine.* J Biomech 1994; 28: 1023-1035.

ASHER M, DESMET A, WHITNEY W, ET AL. (1987). *Changes in spinal alignment resulting from spinal deformity orthoses in patients with adolescent idiopathic scoliosis.* Orthopedic Trans. 1987; 11:105-106.

AUBIN CE, DANSEREAU J, LABELLE M. (1993). *Biomechanical simulation of the effect of the Boston brace on a model of the scoliotic spine and thorax (in French).* Ann Chir 1993; 47-9: 881-887.

AUBIN CE, DESCRIMES JL, DANSEREAU J, ET AL. (1995). *Geometrical modeling of the spine and thorax for biomechanical analysis of scoliotic deformities using finite element method (in French).* Ann Chir 1995; 49: 749-61.

AUBIN CE, DANSEREAU J, ET AL. (1997). *Rib cage spine coupling patterns involved in brace treatment of adolescent idiopathic scoliosis.* Spine 1997; 22:629-635.

AUBIN CE, STOKES I.A.F, FARDNER-MORSE M. (1997). *Biomechanical simulations of Cortel-Dubousset instrumentation maneuvers.* Canadian Orthopedic Society, Hamilton, Canada, May 1997, p.44.

AUBIN CE, DANSEREAU J, ET AL (1997). *Morphometric evaluations of personalized 3D reconstructions and geometric models of the human spine.* Med & Biol Eng & Comput 1997; 35:1-8.

AUBIN CE, DANSEREAU J, PETIT Y, ET AL. (1998). *Three-dimensional measurement of wedged scoliotic vertebrae and intervertebral disks.* Eur Spine J 1998; 7:59-65.

AUBIN CE. (1999). *The scientific approach to treatment is better than the empirical ("trial and error") approach.* Research into Spinal Deformities 2, I.A.F. Stokes (Ed.) IOS Press. 1999.

BAJAJ C.L, BERNARDINI F, ET AL. (1995). *Automatic Reconstruction of Surfaces and Scalar Fields From 3D scans.* Proc. SIGGRAPH'95, ACM, 1995.

BEASTON R.K, CHERRIE J.B, ET AL (1999). *Fast Fitting of radial basis functions: Methods based on preconditioned GMRES iteration.* Advances in Computational Mathematics, 11:253-270, 1999.

BEASTON R.K, CHERRIE J.B, ET AL (2001). *Fast evaluation of radial basis functions: Methods for four-dimensional polyharmonic splines.* SIAM J, Math. Anal., 32(6):1272-1310, 2001.

BERNARDINI F, MITTLEMAN J, ET AL. (1999). *The Ball-Pivoting Algorithm for Surface Reconstruction.* IEEE Transactions on Visualization and Computer Graphics, October-December, 1999.

BLOOMENTHAL J. (1997). *Introduction to Implicit Surfaces.* Morgan Kaufmann, San Francisco, California, 1997.

BOOKSTEIN F.L. (1989). *Principal warps: Thin-plate splines and the decomposition of deformations.* IEEE Trans. Pattern Anal. Machine. Intell., vol. 11, pp. 567-585, June 1989.

BRENT R. (1973). *Algorithms for minimization without derivatives.* Prentice-Hall (1973), republished by Dover in paperback (2002), ISBN 0-486-41998-3.

BRUNSMAN M.A, DAANEN H.M, ET AL. (1997) *Optimal Postures and Positioning for Human Body Scanning.* International Conference on Recent Advances in 3-D Digital Imaging and Modeling, May 12 - 15, 1997

CARR J.C, BEASTON R.K, CHERRIE J.B, ET AL (2001) *Reconstruction and Representation of 3D Objects with Radial Basis Functions.* Proc. SIGGRAPH 2001, ACM, 2001.

DANSEREAU J, STOKES IAF. (1988). *Radiographic reconstruction of 3D human rib cage.* J Biochech 21: 893-901.

DANSEREAU J, LABELLE H, AUBIN C.E. (1993). *3-D personalized parametric modelling of reconstructed scoliotic spines.* IV International Symposium on Computer Simulation in Biomechanics, Paris,France.

DAVIS J, MARSCHNER S.R, GARR M, LEVOY M. (2002). *Filling Holes in Complex Surfaces using Volumetric Diffusion.* 3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on.

DAWSON EG, KROPF MA, PURCELL G, KABL JM, ET AL. (1993). *Optoelectronic evaluation of trunk deformity in scoliosis.* Spine, 18(3):326-331.

DELAUNAY N.B. (1934). *Sur la Sphere.*

DELORME S., PETIT Y., ET AL. (1999) *Three-Dimensional Modelling and Rendering of the Human Skeletal Trunk from 2D Radiographic Images*

DELPHINE P, AUBIN CE, ET AL. (2003). *Boston Brace Correction in Idiopathic Scoliosis: A Biomechanical Study.* Spine 28(15):1672-1677.

DINH H, TURK G, ET AL. (2001). *Reconstructing Surfaces Using Anisotropic Basis Functions.* Proc. ICCV 2001.

EBERLY, DAVID H. (2001). *3D Game Engine Design, A practical approach to real-time computer graphics.* Page 49.

EDELSBRUNNER H, MUCKE E.P. (1992). *Three-Dimensional Alpha Shapes.* ACM Transactions on Graphics, Vol. 13, No. 1, 1994.

EVANS A.C, DAI W, COLLINS L, ET AL. (1991). *Warping of a computerized 3-D atlas to match brain image volumes for quantitative neuroanatomical and functional analysis.* in Proc. SPIE 1445 Medical Imaging V: Image Processing, M.H.Loew, Ed., San Jose, CA, 1991, pp. 236-246.

FASSHAUER G.E. (2003). *Meshfree Method.* Department of Applied Mathematics, Illinoise Institute of Technology, Chicago. http://amadeus.math.iit.edu/ fass/MeshfreeNano.pdf

FARFIELD TECHNOLOGY. (2003). *Manual for FastRBF Toolbox, Command Line Interface, Version 1.4* http://www.fastfield.com.

FORTIN D, CHERIET F, BEAUSEJOUR M, JONCAS J, LABELLE H. (2002). *A 3D Visulization Tool For The Design and Customization of spinal braces.*

FRANKE R. (1982). *Scattered Data Interpolation: Tests of Some Methods.* Mathematics of Computation, Vol. 38 No. 157, pp181-200, January, 1982.

GALANTE J, SCHULTZ A, DEWALD RL, RAY RD. (1970). *Forces acting in the Milwaukee brace on patients undergoing treatment for idiopathic scoliosis.* J Bone Joint Surg Am 1970; 52A: 498-506.

GALASSI M, DAVIES J. ET AL (2003). *GNU Scientific Library—Reference Manual.* Edition 1.4, for GSL Version 1.4. http://www.gnu.org/software/gsl/manual/gsl-ref_toc.html .

GARCEAU P., BEAUSEJOUR M., CHERIET F., ET AL. (2002). *Investigation of Muscle Recruitment Patterns in Scoliosis Using a Biomechanical Finite Element Model.* Research into Spinal Deformities 3, A.Tanguy and B.Peuchot (Eds.) IOS Press, 2002

GEORGE P.L, HECHT F, ET AL. (1991). *Automatic Mesh Generator with Specified Boundary.* Computer Methods in Applied Mechanics and Engineering , North-Holland, vol 92, 269-288.

GIGNAC D, AUBIN CE, DANSEREAU J, ET AL. (2000). *Optimization method for 3D bracing correction of scoliosis using a finite element model.* Eur Spine J 2000; 9: 185-90.

GUISE J.A, MARTEL Y. (1988). *3D-biomedical modeling: mergin image processing and computer aided design.* Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, New Orleans, 1988.

HOPPE H, DEROSE T, ET AL. (1992). *Surface reconstruction from unorganized points.* Proc. SIGGRAPH'92, ACM 1992.

HORN B.K.P (1987). *Closed-form solution of absolute orientation using unit quaternions.* J. opt. Soc. Amer. A vol. 4, no. 4, pp. 629-642, Apr. 1987.

JAREMKO JL. (2001). *Estimation of scoliosis severity from the torso surface by neural networks.* PhD thesis. Univeristy of Calgary.

JIANG DI. (2003). *Visualization and prediction of spatial deformation using thin-plate splines in the context of scoliosis.* Master thesis, Department of Computer Science, University of Montreal.

JOE B. (1991). *GEOMPACK—A Software Package for the Generation of Meshes Using Geometric Algorithms.* Advances in Engineering Software, vol 56, no. 13, pp.325-331.

KEYSER J, KRISHNAN S, ET AL. (1997). *Efficient and Accurate B-rep Generation of Low Degree Sculptured Solids using Exact Arithmetic.* Proceedings of ACM Solid Modeling '97, 1997.

KRIGE DG. (1951). *A statistical approach to some basic mine valuation problems on the Witwatersrand.* J Chem Metall Min Soc S Afr, 1951; 52:119-139.

KRISHNAN S, MANOCHA D. (1997). *An Efficient Surface Intersection Algorithm based on Lower Dimensional Formulation.* ACM Trans. on Computer Graphics, 16:74-106, 1997.

KUNIS S, POTTS D, STEIDL G, (2002). *NFFT, Software package (C-library).* University Lubeck, http://www.math.uni-luebeck.de/potts/nfft/, 2002

KUNIS S, POTTS D, STEIDL G, (2002). *Fast Fourier transforms at nonequispaced knots: A user's guide to a C-library.* University Lubeck, http://www.math.uni-luebeck.de/potts/nfft/, 2002

LABELLE H, DANSEREAU J, BELLEFLEUR C, POITRAS B. (1996). *3-D effect of the Boston brace on the thoracic spine and rib-cage.* Spine 1996; 21(1)59-64.

LAWSON C.L. (1977). *Software for C1 Surface Interpolation.* Mathematical Software III, pp.161-175.

LO S.H. (1991). *Volume Discretization into Tetrahedra - I. Verification and Orientation of Boundary Surfaces.* Computers and Structures, vol 39, no. 5, pp.493-500

LO S.H. (1991). *Volume Discretization into Tetrahedra - II. 3D Triangulation by Advancing Front Approach.* Computers and Structures, vol 39, no. 5, pp.501-511

LOHNER R, PARIKH P, ET AL (1988). *Interactive Generation of Unstructed Grid for Three Dimensional Problems.* Numberical Grid Generation in Computational Fluid Mechanics '88, Pineridge Press, pp. 687-500

LOHNER R. (1996). *Progress in Grid Generation via the Advancing Front Technique.* Engineering with Computers, vol 12, pp.186-210

LORENSEN W.E, CLINE H.E. (1987). *Marching cubes: A high resolution 3D surface construction algorithm.* Computer Graphics, 21(4):163-169, July 1987.

LUEBKE D.P. (2001). *A Developer's Survey of Polygonal Simplification Algorithms.* IEEE Computer Graphics and Application. May/June 2001.

MARRAS WS, SOMMERICH CM. (1991). *A three-dimensional motion model of loads on the lumbar spine: I. model structure.* Hum Factors 1991; 33: 123-37.

MCGLINE C. (1989). *Analytic Data reduction Schemes in Non-Topographic Photogrammetry.* H.M. Karara, American Society for Photogrammetry and Remote Sensing, 37-57, 1989.

MILENKOVIC V.J. (1993). *Robust Polygon Modeling.* Computer-Aided Design, 25:546-566, September 1993.

NACHEMSON A, PETERSON L. (1995). *Effectiveness of treatment with a brace in girls who have adolescent idiopathic scoliosis.A prospective controlled study based on data from the brace study of the scoliosis research society.* The Journal of Bone and Joint Surgery 1995; 77-A(6):815-821.

NIESLONY A, POTTS D, STEIDL G. (2003). *Rapid evaluation of radial functions by fast Fourier transforms at nonequispaced knots.* University Mannheim, preprint 2003.

National Scoliosis Foundation *http://www.scoliosis.org.*

OGILVIE J. (1994). *Spinal orthotics. In: An Overview, the Pediatric Spine: Principles and Practice.* S.L. Weinstein, ed. New York, NY: Raven; 1994:1787-93.

OWEN S, (2003). *A Survey of Unstructured Mesh Generation Technology.* http://www.andrew.cmu.edu/user/sowen/. June, 2003.

PONCET P, DELORME S. ET AL. (1999). *3D Reconstructions of the External and Internal Geometries of the Trunk Using Laser and Stereo-Radiographic Imaging Techniques.* Research into Spinal Deformities 2, I.A.F. Stokes (Ed.) IOS Press, 1999.

POTTS D, STEIDL G. (2003). *Fast summation at nonequispaced knots by NFFTs.* SIAM J. Sci. Comput. 24 (2003), 2013 - 2037.

ROBERTS SB, CHEN PH. (1989). *Elastostatic analysis of the human thoracic skeleton. J Biomech* 1970; 3:527-545.

ROBERTS SB, CHEN PH. (1989). *Global geometric characteristics of typical human ribs. J Biomech* 1972; 5:191-201.

ROHR K. STIEHL H.S. ET AL. (2001). *Landmark-Based Elastic Registration Using Approximating Thin-Plate Splines.* IEEE Transactions on medical imaging, Vol. 20, No.6 June 2001.

ROY M, BOUTARD A, LABELLE H. (1996). *Can digital radiography decrease radiation exposure for adolescents with idiopathic scoliosis?* 26^{th} Annual Meeting of the Quebec Scoliosis Society 1996, Hull.

SCHOBERL J. (1994). http://www.hpfem.jku.at/netgen.

SCHOBERL J. (2004). *Manual for NERTGEN–4.3.* http://www.hpfem.jku.at/netgen.

SCHROEDER J.W, ZARGE A.J, LORENSEN E.W. (1992) *Decimation of Triangle Meshes.* Computer Graphics, 26, 2, July 1992.

SHEPHARD S.M, GEORGES K.M. (1991). *Three-Dimensional Mesh Generation by Finite Octree Technique.* International Journal for Numberical Methods in Engineering, vol 32, pp. 709-749.

SCUTT NF, DANGERFILD PH, DORGAN JC. (1996). *The relationship between surface and radiological deformity in adolescent idiopathic scoliosis.* Eur Spine J, 5(2): 85-90.

STOKES IAF, BIGALOW LC, AND MORELAND MS. (1987). *Three-dimensional spinal curvature in idiopathic scoliosis.* J Orthop Res 5: 102-113, 1987,

STOKES IAF, DANSEREAU J, AND MORELAND MS. (1989). *Rib cage asymmetry in idiopathic scoliosis.* Journal of Orthopedic Research, 7, 599-606.

SUNDARAM SH, FENG CC. (1977). *Finite element analysis of the human thorax.* J Biomech 1977; 10: 505-16.

TREECE G.M, PRAGER R.W, ET AL. (1999). *Regularised marching tetrahedra: improved iso-surface extraction.* Computers and Graphics, 23(4):583-598, 1999.

TROCHU F. (1993). *A contouring program based on dual krigin interpolation.* Engineering with computers, 1993; 9: 160-177.

TURNER-SMITH AR, HARRIS JD, HOUGHTON GR, JEFFERSON RJ. (1998). *A method for analysis of back shape in scoliosis.* J Biomech, 21(6):497-509.

UEYOSHI A, SHIMA Y. (1985). *Studies on spinal braces, with special reference to the effects on increased abdominal pressure.* Int Orthop 1985; 9: 255-8.

VICTOR O, CHARLES D, PIERRE-MARC J. (2004). *Fast Hierarchical Importance Sampling with Blue Noise Properties.* University of Montreal. To appear in the proceedings of ACM SIGGRAPH 2004.

WAHBA G. (1990). *Spline Models for Observational Data.* Philadelphia, PA; Soc. Ind. Appl. Math., 1990.

WANG Y. (1998). *Smoothing spline models with correlated random errors.* J. Amer. Statist, Assoc., Vol. 93, no. 441 pp. 341-348, 1998.

WATSON F.D. (1981). *Computing the Delaunay Tesselation with Application to Voronoi Polytopes.* The Computer Journal, Vol 24(2) pp.167-172.

WEATHERILL N.P. HASSAN O. (1994). *Automatic Mesh Generator with Specified Boundary.* International Journal for Numberical Methods in Engineering, vol 37, pp.2005-2039.

WEINSTEIN, S.L. *Idiopathic scoliosis: Natural history of curve progression.* Proceedings of the Scoliosis Research Society. 71-78: 1984.

WHITAKER R. (1998). *A Level-set Approach to 3D Reconstruction from range data.* International journal of Computer Vision, Vol. 29, No. 3, October, 1998.

YERRY A.M, SHEPHARD S.M. (1984). *Three-Dimensional Mesh Generation by Modified Octree Technique.* International Journal for Numerical Methods in Engineering, vol 20, pp.1965-1990.

APPENDIX I

REGISTRATION RESULT

I.1 Estimate Minimum Interval

In this appendix we list the experiment results of using brute force method to estimate the minimum interval for tps_{ap} transformation. Table I.1 is a simple version of our results, the external trunk skin models are from 5 patients. We calcuate the $BPTSMD$ values of using different λ. The range of λ is about $(-10486, 10486)$ (tps_{ap} linear system equation is stable when λ approachs $\pm\infty$), and λ is chosen with the following logarithm equation:

$$\lambda = 1 + sign(i) * c * 2_{abs(i)/k}$$

Where $i = -20, -19, \ldots, 20$, and $c = 0.01, k = 1$. The tps_{ap} linear system equation is ill-conditioned when λ is in region $(0.9, 20)$, and the optimal value often appears in $(-3, 1)$. So more λ values are chosen from this region.

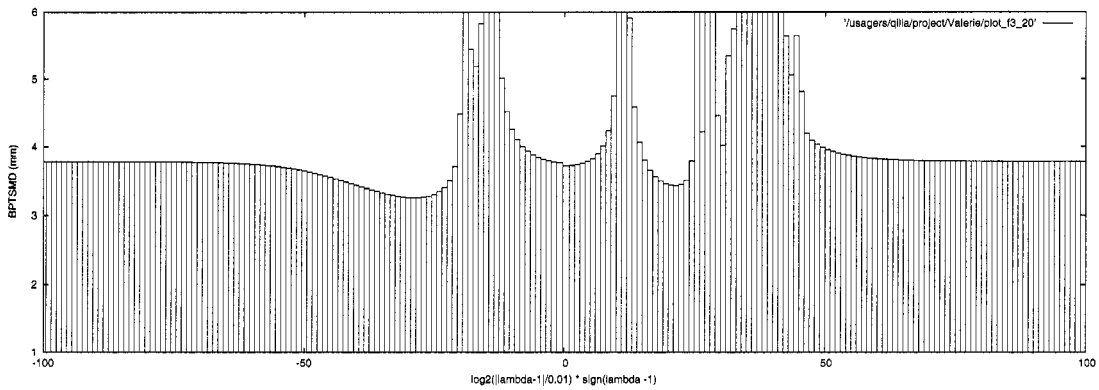
In Figure I.1, I.2, I.3, I.4 and I.5, we draw the curves of $BPTSMD$ vs λ. The λ values are chosen with the same logarithm equation as above. But this time, we let $i = -100, -99, \ldots, 100$, and $c = 0.01, k = 5$. We only shows results from 4 patients. More experiment results are not displayed in this thesis.

I.2 Evaluate Registration Result

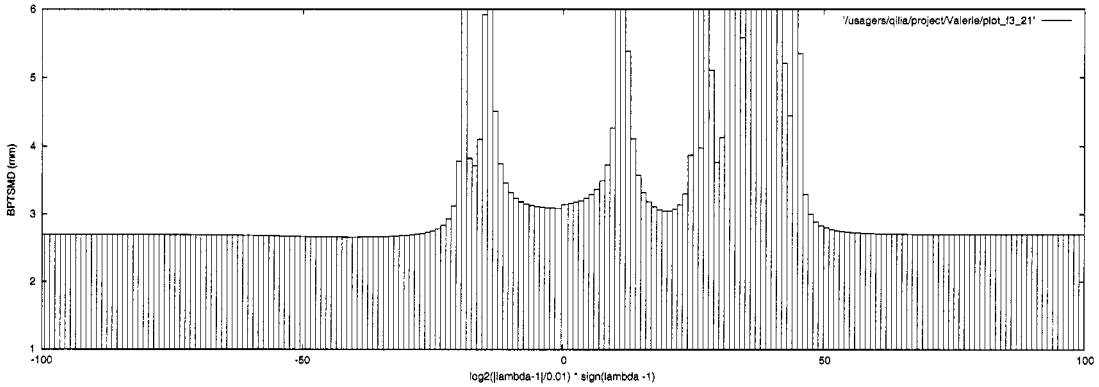
Table I.2, I.3 and I.4 show experiment results the different registration methods ($rigid$, $affine$, tps_{in}, $tps_{in}(r^3)$, tps_{ap}) . And the optimal tradeoff parameter for tps_{ap} transformation is also computed.

Table I.1 Estimate the minimum interval of optimal tradeoff parameter using brute force method to calcuate the *BPTSMD* vs λ. The external models are from 5 patients. "$3 \sim 1$" (in second row and second column) means that the source model is $F00030003$, and target model is $F00030001$. λ is computed by logarithm equation: $\lambda = 1 + sign(i) * c * 2^{abs(i)/k}$, where $i \in \{-20, -19, \ldots, 20\}$, and $c = 0.01, k = 1$.

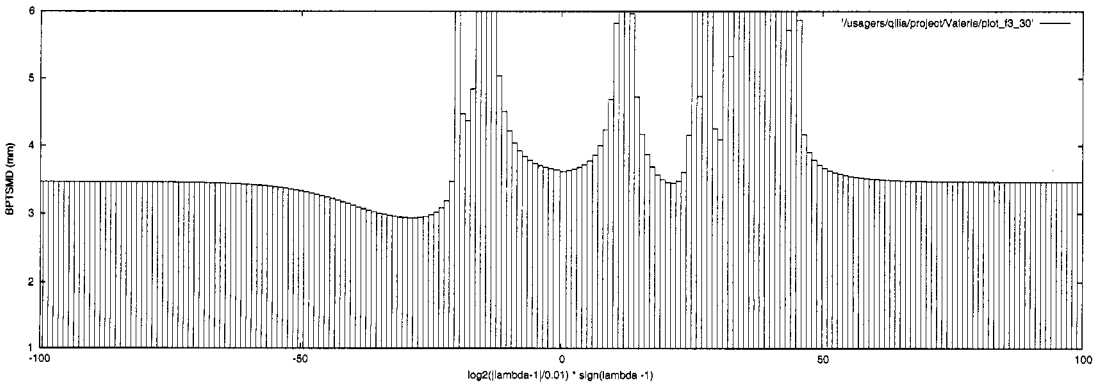
λ	$F0003$		$F0006$		$F0025$		$F0025$		$F0027$	
	$3 \sim 1$	$2 \sim 0$	$3 \sim 1$	$2 \sim 0$	$3 \sim 1$	$2 \sim 0$	$3 \sim 1$	$2 \sim 0$	$3 \sim 1$	$2 \sim 0$
-104846	2.4772	3.7832	3.6574	3.2435	7.3013	6.3703	3.6936	4.3039	4.0259	4.5205
-5241	2.4771	3.7831	3.6573	3.2432	7.2988	6.3685	3.6932	4.3034	4.0255	4.5200
-2620	2.4769	3.7828	3.6569	3.2425	7.2941	6.3649	3.6926	4.3025	4.0249	4.5191
-1309	2.4766	3.7822	3.6563	3.2411	7.2848	6.3578	3.6913	4.3008	4.0237	4.5173
-654.3	2.4760	3.7811	3.6550	3.2384	7.2663	6.3437	3.6886	4.2972	4.0211	4.5137
-326.6	2.4747	3.7789	3.6524	3.2331	7.2298	6.3158	3.6835	4.2904	4.0160	4.5064
-162.8	2.4723	3.7746	3.6475	3.2226	7.1585	6.2613	3.6732	4.2767	4.0059	4.4923
-80.92	2.4677	3.7661	3.6381	3.2032	7.0229	6.1579	3.6533	4.2513	3.9863	4.4653
-39.96	2.4597	3.7496	3.6205	3.1676	6.7757	5.9682	3.6159	4.2042	3.9498	4.4154
-19.48	2.4469	3.7185	3.5892	3.1064	6.3674	5.6540	3.5499	4.1201	3.8863	4.3299
-9.24	2.4303	3.6632	3.5383	3.0105	5.7717	5.1916	3.4421	3.9930	3.7855	4.2002
-4.12	2.4144	3.5746	3.4645	2.8894	5.0505	4.6291	3.2894	3.8256	3.6497	4.0360
-1.56	2.4066	3.4617	3.3634	2.7743	4.3307	4.0983	3.1153	3.6728	3.4989	3.8691
-0.28	2.4162	3.3471	3.2374	2.7055	3.8293	3.7220	2.9645	3.6006	3.3584	3.7210
0.36	2.4491	3.2653	3.2377	2.7852	3.6068	3.4942	2.8985	3.6927	3.2857	3.6230
0.68	2.5097	3.2804	13.881	13.109	3.6070	3.4350	2.9809	3.8602	6.9229	8.1762
0.84	2.9857	3.7101	7.5182	6.1221	3.7151	3.4622	3.1753	4.3800	3.6845	3.7315
0.92	4.6768	5.8272	14.927	8.6012	3.8551	3.5167	3.4060	4.3310	3.9438	3.8401
0.96	3.6323	4.5156	13.026	15.861	3.9768	3.5676	3.6296	4.6983	4.1929	3.9685
0.98	3.1667	3.8828	19.353	38.942	4.0607	3.6044	3.7931	5.1755	4.3730	4.0724
0.99	3.0926	3.7646	35.241	110.96	4.1106	3.6273	3.8952	5.5845	4.4841	4.1396
1.01	3.1259	3.7187	24.252	17.638	4.2311	3.6856	4.1581	7.1226	4.7648	4.3192
1.02	3.2859	3.8282	16.934	13.884	4.3046	3.7236	4.3304	8.6443	4.9444	4.4400
1.04	5.2550	6.4460	12.893	11.312	4.4895	3.8288	4.8116	10.796	5.4245	4.7830
1.08	3.5547	3.8017	18.420	18.783	5.1664	4.2855	7.2760	6.0374	7.4095	6.8735
1.16	3.0277	3.4380	21.215	9.8577	10.158	28.357	9.2025	32.479	12.976	12.270
1.32	7.7596	13.921	9.5865	6.7968	8.9378	6.1571	6.1035	6.0071	8.2450	7.6014
1.64	4.2568	4.0152	21.623	67.607	7.8174	7.9274	7.0539	15.392	17.708	10.285
2.28	16.715	25.595	50.443	25.236	19.802	28.629	13.675	19.990	14.618	14.845
3.56	53.574	46.085	6.1838	4.8674	25.544	24.708	27.209	24.772	25.476	28.522
6.12	6.9262	4.8162	30.065	52.894	33.257	30.574	25.417	38.465	11.868	16.561
11.24	2.6488	3.9554	3.9504	3.9526	13.006	10.695	4.2317	5.1077	4.6346	5.4564
21.48	2.5347	3.8605	3.7583	3.4715	9.0012	7.6563	3.8942	4.5892	4.2386	4.8324
41.96	2.5015	3.8201	3.7020	3.3419	8.0103	6.9051	3.7852	4.4308	4.1200	4.6546
82.92	2.4884	3.8013	3.6788	3.2899	7.6282	6.6182	3.7373	4.3640	4.0704	4.5834
164.84	2.4826	3.7922	3.6680	3.2662	7.4597	6.4908	3.7149	4.3336	4.0477	4.5513
328.68	2.4799	3.7878	3.6626	3.2548	7.3802	6.4303	3.7043	4.3186	4.0367	4.5358
656.36	2.4785	3.7855	3.6601	3.2493	7.3414	6.4011	3.6992	4.3112	4.0314	4.5284
1311.7	2.4779	3.7844	3.6588	3.2465	7.3225	6.3865	3.6966	4.3080	4.0287	4.5246
2622.4	2.4775	3.7839	3.6582	3.2452	7.3131	6.3792	3.6953	4.3062	4.0274	4.5228
5243.8	2.4774	3.7836	3.6579	3.2445	7.3084	6.3756	3.6946	4.3053	4.0268	4.5219
10486	2.4773	3.7835	3.6577	3.2442	7.3060	6.3738	3.6942	4.3048	4.0264	4.5214



(a) Source model: $F00030002$; Target model: $F00030000$.

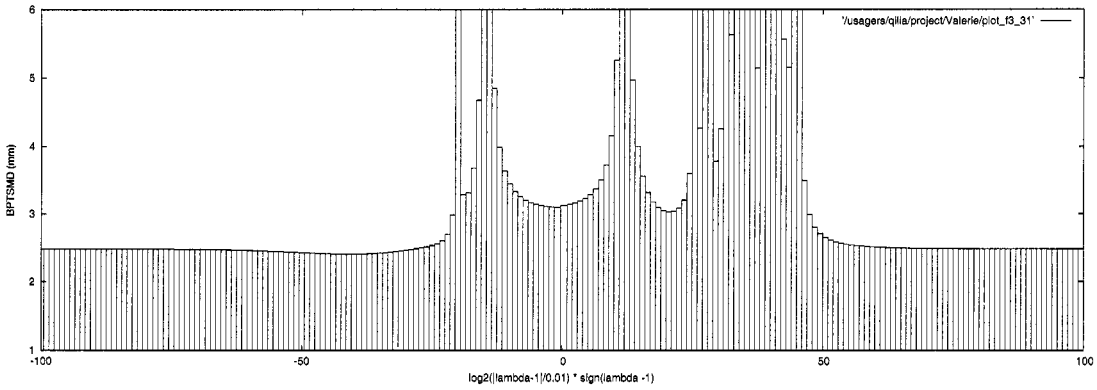


(b) Source model: $F00030002$; Target model: $F00030001$.

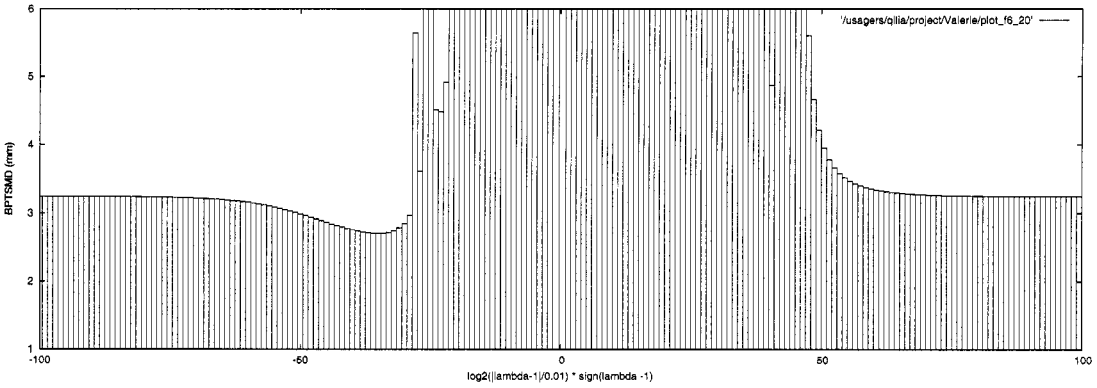


(c) Source model: $F00030003$; Target model: $F00030000$.

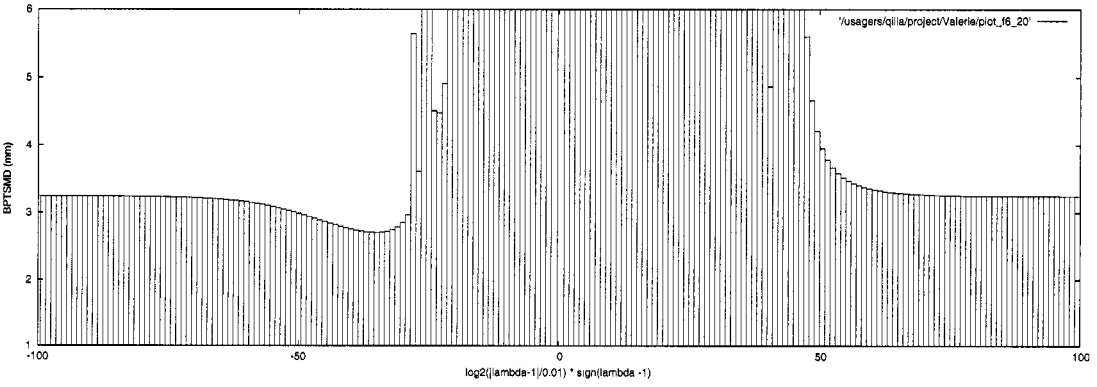
Figure I.1 Estimate the minimum interval of optimal tradeoff parameter: y-axis is *BPTSMD* (mm). x-axis is i, the λ values are computed by logarithm equation: $\lambda = 1 + sign(i) * c * 2^{abs(i)/k}$, where $i = -100, -99, \ldots, 100, c = 0.01, k = 5$.



(a) Source model: $F00030003$; Target model: $F00030001$.

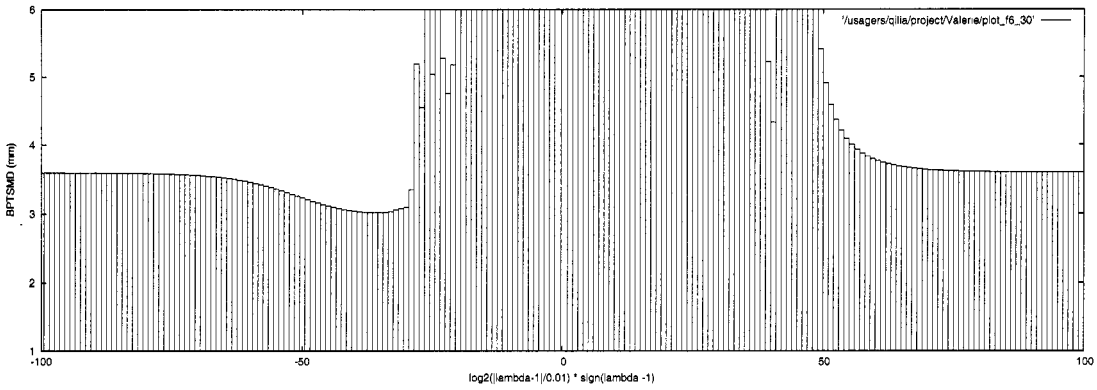


(b) Source model: $F00060002$; Target model: $F00060000$.



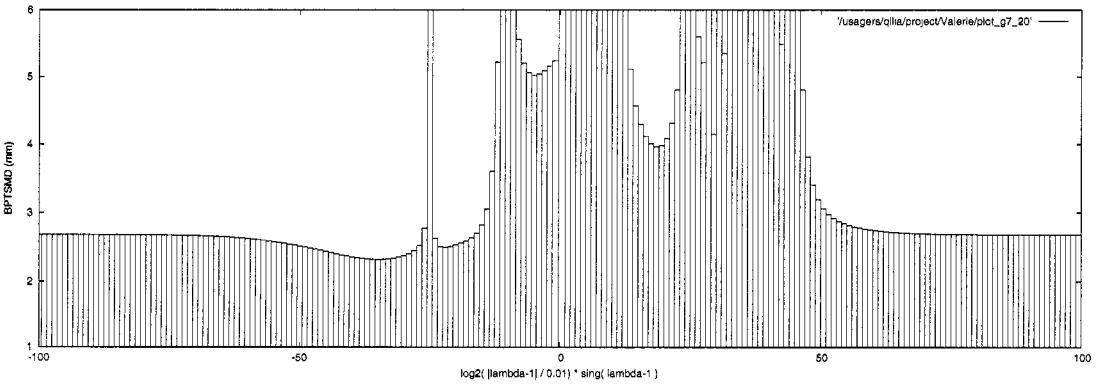
(c) Source model: $F00060002$; Target model: $F00060001$.

Figure I.2 (Continued) Estimate the minimum interval of optimal tradeoff parameter.



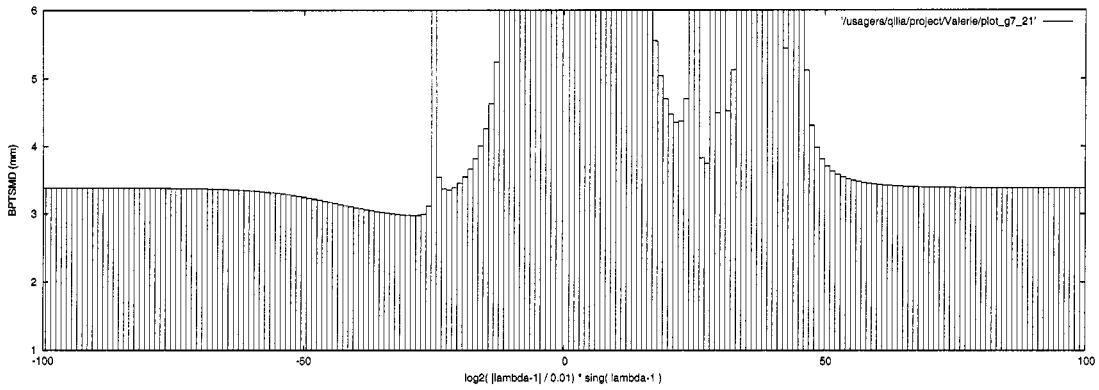
(a) Source model: $F00060003$; Target model: $F00030000$.

(b) Source model: $F00060003$; Target model: $F00060001$.

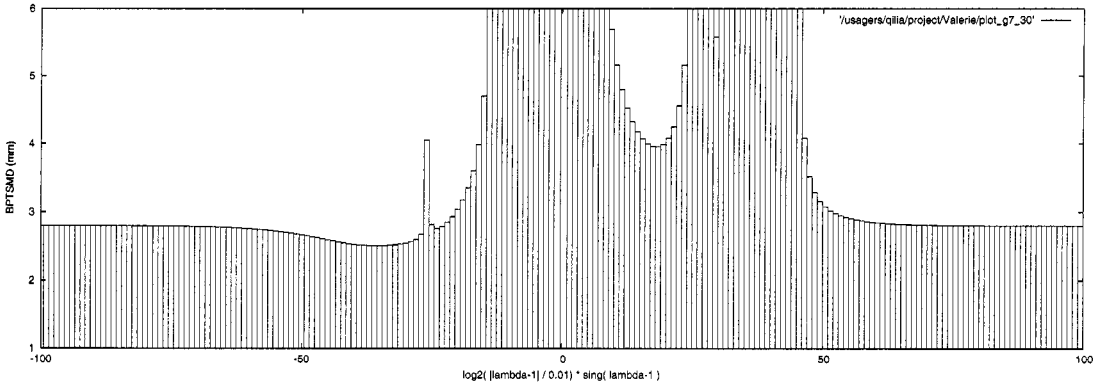


(c) Source model: $G00070002$; Target model: $G00070000$.

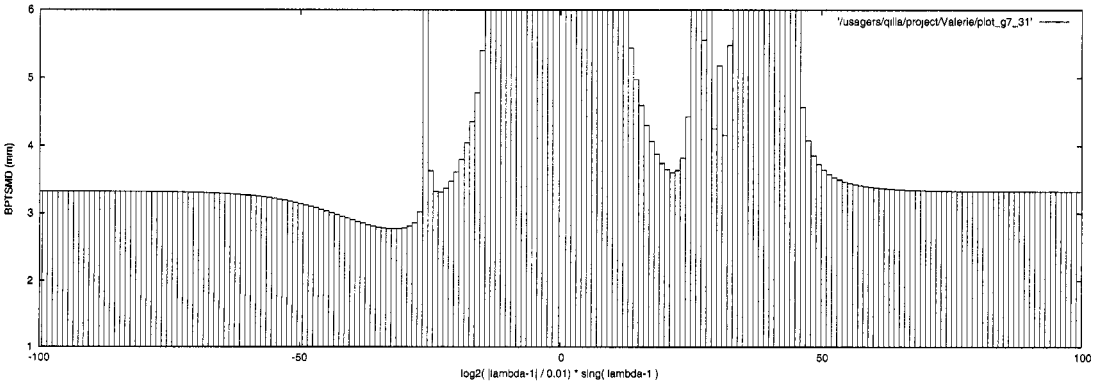
Figure I.3 (Continued) Estimate the minimum interval of optimal tradeoff parameter.



(a) Source model: $G00070002$; Target model: $G00070001$.

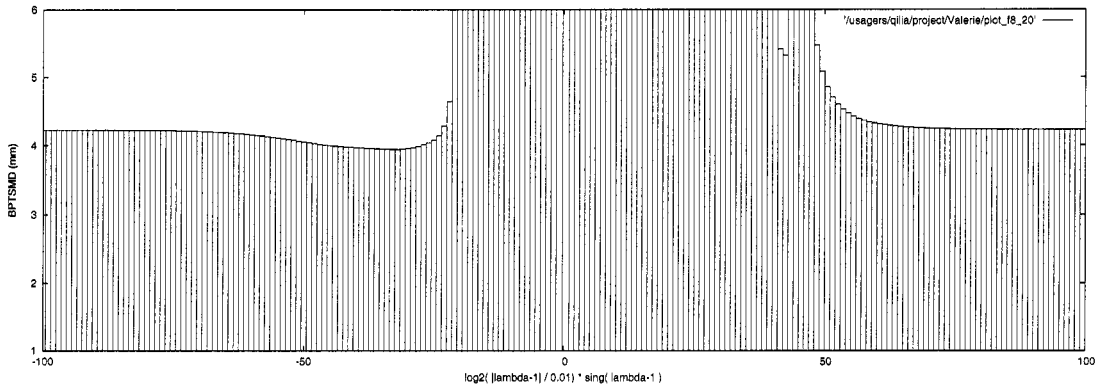


(b) Source model: $G00070003$; Target model: $G00070000$.

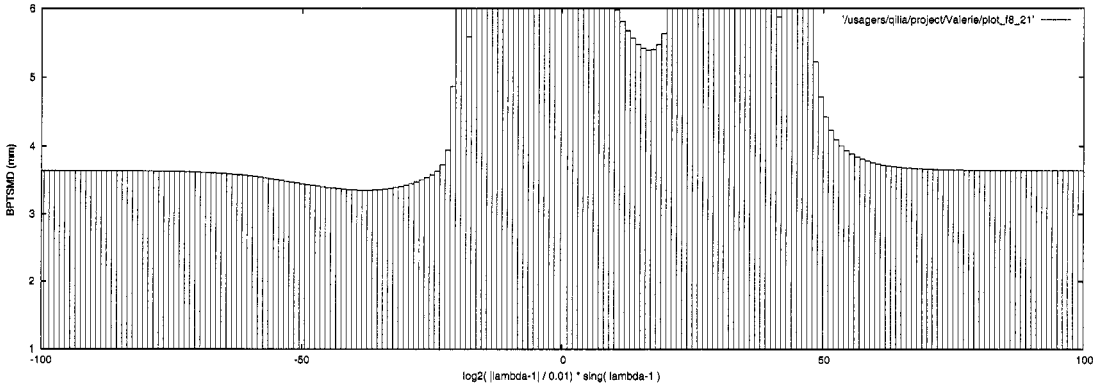


(c) Source model: $G00070003$; Target model: $G00070001$.

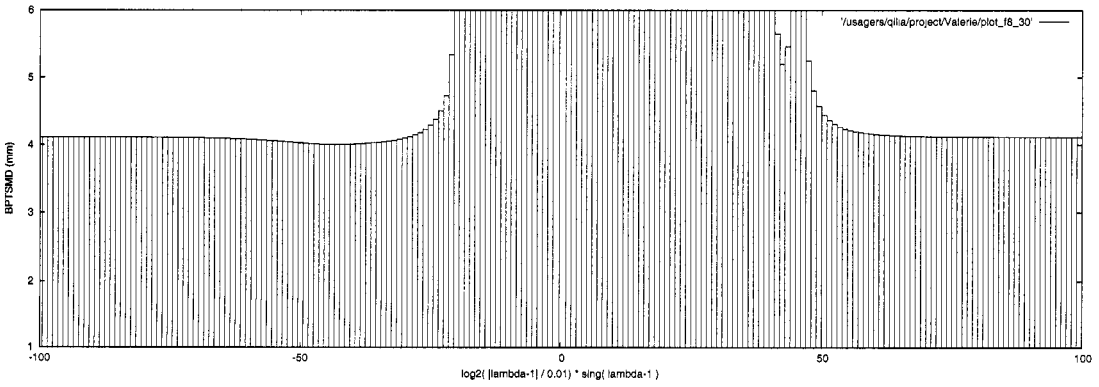
Figure I.4 (Continued) Estimate the minimum interval of optimal tradeoff parameter.



(a) Source model: $F00080002$; Target model: $F00080000$.



(b) Source model: $F00080002$; Target model: $F00080001$.



(c) Source model: $F00080003$; Target model: $F00080000$.

Figure I.5 (Continued) Estimate the minimum interval of optimal tradeoff parameter.

Table I.2 Result of the different registration methods. patients: F0003, F0006, G0007, F0008; **Bidirectional Point-to-Surface Mean Distance** ($BPTSMD, mm$); pID: ID of the patient; src: source model (posture); dst: target model (posture); tps_{in}: Thin-Plate Spline Interpolation ($m = 2, d = 3$, kernel: R); $tps_{in}(r^3)$: Thin-Plate Spline Interpolation ($m = 3, d = 3$, kernel: R^3); tps_{ap}: Thin-Plate Spine Approximation ($m = 2, d = 3$, kernel: R); λ: optimal tradeoff parameter of tps_{ap}.

pID	src	dst	$rigid$	$\frac{rigid}{tps_{ap}}$	$affine$	$\frac{affine}{tps_{ap}}$	tps_{in}	$\frac{tps_{in}}{tps_{ap}}$	$tps_{in}(r^3)$	$\frac{tps_{in}(r^3)}{tps_{ap}}$	tps_{ap}	λ
F0003	0	1	2.252	1.405	2.011	1.255	1.634	1.019	4.1	2.558	1.603	0.54
F0003	0	2	4.01	1.223	3.846	1.174	3.343	1.02	2.643	0.806	3.277	0.53
F0003	0	3	3.645	1.245	3.484	1.19	2.993	1.022	2.566	0.876	2.928	0.54
F0003	1	0	2.256	1.395	2.023	1.251	1.647	1.019	3.044	1.883	1.617	0.54
F0003	1	2	3.213	1.183	2.772	1.02	2.72	1.001	4.212	1.55	2.717	-0.56
F0003	1	3	2.807	1.145	2.521	1.028	2.461	1.004	3.846	1.569	2.452	-0.95
F0003	2	0	4.022	1.236	3.783	1.162	3.312	1.017	2.67	0.82	3.255	0.52
F0003	2	1	3.203	1.204	2.698	1.015	2.667	1.003	5.259	1.978	2.659	-1.54
F0003	2	3	0.841	1.07	0.945	1.202	0.803	1.021	0.832	1.059	0.786	0.67
F0003	3	1	2.814	1.169	2.477	1.029	2.425	1.008	4.98	2.069	2.407	-1.55
F0003	3	2	0.835	1.061	0.95	1.207	0.805	1.023	0.831	1.056	0.787	0.67
F0006	0	1	3.353	2.402	2.439	1.747	1.422	1.019	2.643	1.894	1.396	0.22
F0006	0	2	2.743	1.009	3.199	1.177	2.72	1.001	4.014	1.476	2.719	-0.13
F0006	0	3	2.883	0.931	3.623	1.17	3.114	1.005	3.96	1.279	3.097	-0.57
F0006	1	0	3.362	2.525	2.353	1.767	1.356	1.018	2.631	1.976	1.332	0.22
F0006	1	2	5.2	1.707	4.143	1.359	3.05	1.001	5.507	1.807	3.047	0.07
F0006	1	3	3.934	1.256	3.527	1.126	3.134	1	4.453	1.422	3.133	0.07
F0006	2	0	2.712	1.003	3.244	1.2	2.705	1	4.144	1.533	2.704	-0.1
F0006	2	1	5.116	1.611	4.372	1.377	3.187	1.004	5.432	1.711	3.175	0.12
F0006	2	3	2.252	1.261	2.505	1.403	1.786	1	2.714	1.52	1.785	-0.05
F0006	3	0	2.847	0.944	3.595	1.192	3.02	1.001	3.676	1.218	3.017	-0.16
F0006	3	1	3.896	1.221	3.658	1.147	3.2	1.003	4.351	1.364	3.19	0.15
F0006	3	2	2.254	1.328	2.454	1.446	1.697	1	2.852	1.681	1.697	-0.01
G0007	0	1	2.092	1.2	1.982	1.137	1.75	1.004	2.006	1.151	1.743	0.26
G0007	0	2	3.2	1.358	2.697	1.144	2.361	1.002	2.88	1.222	2.357	-0.23
G0007	0	3	2.995	1.197	2.793	1.117	2.506	1.002	3.074	1.229	2.501	-0.39
G0007	1	0	2.097	1.201	1.969	1.128	1.747	1	2.031	1.163	1.746	0.1
G0007	1	2	3.155	1.076	3.361	1.146	2.987	1.018	3.235	1.103	2.933	0.68
G0007	1	3	3.233	1.192	3.258	1.201	2.724	1.005	3.268	1.205	2.712	0.26
G0007	2	0	3.189	1.372	2.683	1.155	2.325	1.001	2.749	1.183	2.324	-0.12
G0007	2	1	3.164	1.064	3.382	1.137	3.011	1.012	3.126	1.051	2.975	0.54
G0007	2	3	2.199	1.416	1.84	1.185	1.611	1.038	1.582	1.019	1.553	0.56
G0007	3	0	3.013	1.201	2.802	1.116	2.511	1.001	3.054	1.217	2.51	-0.18
G0007	3	1	3.279	1.183	3.325	1.199	2.786	1.005	3.125	1.127	2.772	0.27
G0007	3	2	2.196	1.381	1.871	1.177	1.636	1.029	1.66	1.044	1.59	0.5
F0008	0	1	1.78	1.01	1.778	1.009	1.774	1.006	1.908	1.082	1.763	-3.08
F0008	0	2	4.117	1.061	4.203	1.082	3.921	1.01	4.143	1.067	3.882	0.73
F0008	0	3	4.181	1.033	4.12	1.018	4.047	1	4.436	1.096	4.046	-0.09
F0008	1	0	1.776	1.013	1.771	1.01	1.761	1.005	1.849	1.055	1.753	-1.98
F0008	1	2	3.966	1.203	3.651	1.108	3.297	1	3.829	1.161	3.296	0.06
F0008	1	3	4.1	1.161	3.767	1.066	3.542	1.003	4.103	1.161	3.533	-0.5
F0008	2	0	4.047	1.027	4.224	1.072	3.944	1.001	4.539	1.152	3.941	0.2
F0008	2	1	3.856	1.15	3.642	1.086	3.368	1.005	4.08	1.217	3.353	-0.59
F0008	2	3	1.462	1.099	1.46	1.098	1.366	1.027	1.848	1.39	1.33	-1.36
F0008	3	0	4.117	1.028	4.113	1.027	4.041	1.009	4.816	1.203	4.004	-2.48
F0008	3	1	3.996	1.13	3.759	1.063	3.582	1.013	4.367	1.235	3.537	-1.19
F0008	3	2	1.463	1.099	1.462	1.099	1.365	1.026	1.828	1.374	1.33	-1.34

Table I.3 Result of the different registration methods (Continued). patients: F0014, G0015, F0016, F0017. model $F00140003$ not available.

pID	src	dst	$rigid$	$\frac{rigid}{tps_{ap}}$	$affine$	$\frac{affine}{tps_{ap}}$	tps_{in}	$\frac{tps_{in}}{tps_{ap}}$	$tps_{in}(r^3)$	$\frac{tps_{in}(r^3)}{tps_{ap}}$	tps_{ap}	λ
F0014	0	1	2.413	1.889	1.612	1.261	1.322	1.035	2.425	1.898	1.278	-0.91
F0014	0	2	7.048	2.205	4.281	1.339	3.273	1.024	2.904	0.909	3.196	0.44
F0014	1	0	2.418	1.847	1.641	1.254	1.354	1.034	2.507	1.916	1.309	-0.9
F0014	1	2	5.374	1.419	4.084	1.078	3.792	1.001	3.521	0.93	3.787	-0.26
F0014	2	0	7.061	2.102	4.54	1.352	3.483	1.037	3.043	0.906	3.359	0.48
F0014	2	1	5.369	1.358	4.226	1.069	3.956	1.001	3.654	0.925	3.953	-0.17
G0015	0	1	3.288	1.235	3.168	1.19	2.662	1	3.173	1.192	2.661	0.06
G0015	0	2	6.335	1.685	4.345	1.155	3.824	1.017	5.334	1.418	3.76	-1.11
G0015	0	3	5.199	1.31	4.706	1.186	4.051	1.021	5.435	1.37	3.968	-1.21
G0015	1	0	3.265	1.221	3.178	1.189	2.674	1	3.142	1.176	2.673	0.09
G0015	1	2	6.24	1.502	4.825	1.162	4.188	1.008	5.179	1.247	4.154	-0.75
G0015	1	3	5.609	1.278	5.095	1.161	4.493	1.024	5.664	1.291	4.389	-1.61
G0015	2	0	6.262	1.646	4.673	1.228	3.807	1.001	5.18	1.361	3.805	-0.15
G0015	2	1	6.227	1.462	5.08	1.193	4.265	1.002	5.31	1.247	4.258	-0.28
G0015	2	3	3.699	2.156	2.318	1.351	1.715	1	2.501	1.458	1.715	0.02
G0015	3	0	5.112	1.355	4.928	1.306	3.775	1.001	5.342	1.416	3.773	-0.17
G0015	3	1	5.536	1.283	5.219	1.209	4.37	1.013	5.774	1.338	4.316	-1.1
G0015	3	2	3.692	2.237	2.265	1.372	1.65	1	2.212	1.34	1.65	-0.01
F0016	0	1	1.449	1.002	2.223	1.538	1.48	1.024	1.579	1.092	1.446	0.27
F0016	0	2	2.88	0.957	3.395	1.128	3.01	1	3.75	1.246	3.009	0.08
F0016	0	3	2.482	0.852	2.976	1.021	2.963	1.017	3.61	1.239	2.914	-1.63
F0016	1	0	1.447	1.006	2.172	1.51	1.46	1.015	1.887	1.312	1.439	0.21
F0016	1	2	2.822	0.955	2.987	1.011	3.043	1.03	3.984	1.349	2.953	-4.39
F0016	1	3	2.846	0.882	3.276	1.016	3.267	1.013	3.954	1.226	3.225	-2.91
F0016	2	0	2.868	0.964	3.424	1.15	2.981	1.002	3.672	1.233	2.977	0.12
F0016	2	1	2.805	0.963	2.963	1.018	2.975	1.022	3.962	1.361	2.911	-2.85
F0016	2	3	1.569	1.32	1.394	1.172	1.192	1.002	1.561	1.313	1.189	0.2
F0016	3	0	2.457	0.855	2.959	1.03	2.904	1.011	3.595	1.251	2.873	-0.95
F0016	3	1	2.828	0.892	3.251	1.025	3.183	1.004	3.862	1.218	3.17	-0.76
F0016	3	2	1.569	1.32	1.389	1.168	1.192	1.002	1.445	1.215	1.189	0.2
F0017	0	1	3.826	1.568	2.982	1.222	2.445	1.002	2.925	1.198	2.441	-0.2
F0017	0	2	3.753	1.442	3.769	1.448	2.602	1	3.578	1.375	2.602	0.03
F0017	0	3	3.595	1.714	3.286	1.566	2.101	1.001	3.841	1.831	2.098	-0.1
F0017	1	0	3.836	1.529	3.006	1.198	2.525	1.007	2.922	1.165	2.508	-0.47
F0017	1	2	6.275	1.644	5.506	1.442	3.817	1	4.329	1.134	3.818	0
F0017	1	3	5.169	1.697	4.561	1.498	3.05	1.001	3.827	1.257	3.045	0.1
F0017	2	0	3.586	1.448	3.782	1.527	2.479	1.001	3.584	1.448	2.476	0.17
F0017	2	1	6.155	1.809	5.49	1.614	3.41	1.003	4.251	1.25	3.402	0.21
F0017	2	3	2.552	1.472	2.789	1.609	1.745	1.007	3.225	1.861	1.733	0.19
F0017	3	0	3.483	1.672	3.265	1.568	2.087	1.002	3.943	1.893	2.083	-0.1
F0017	3	1	5.091	1.748	4.455	1.53	2.92	1.003	4.01	1.377	2.912	0.17
F0017	3	2	2.56	1.406	2.671	1.467	1.828	1.004	3.278	1.8	1.821	0.16

Table I.4 Result of the different registration methods (Continued). patients: F0025, F0026, F0027, F0029.

pID	src	dst	$rigid$	$\frac{rigid}{tps_{ap}}$	$affine$	$\frac{affine}{tps_{ap}}$	tps_{in}	$\frac{tps_{in}}{tps_{ap}}$	$tps_{in}(r^3)$	$\frac{tps_{in}(r^3)}{tps_{ap}}$	tps_{ap}	λ
F0025	0	1	2.39	1.669	2.401	1.677	1.437	1.003	1.89	1.319	1.432	0.15
F0025	0	2	7.749	2.098	6.753	1.829	3.839	1.04	3.709	1.004	3.693	0.65
F0025	0	3	7.876	2.239	7.012	1.993	3.787	1.077	3.409	0.969	3.518	0.72
F0025	1	0	2.39	1.64	2.452	1.682	1.465	1.005	1.944	1.334	1.457	0.17
F0025	1	2	8.783	2.35	7.313	1.957	3.796	1.016	4.355	1.166	3.737	0.35
F0025	1	3	8.828	2.333	7.781	2.056	3.89	1.028	4.031	1.065	3.785	0.42
F0025	2	0	7.807	2.279	6.372	1.86	3.623	1.058	3.363	0.982	3.425	0.68
F0025	2	1	8.764	2.5	6.777	1.933	3.602	1.028	3.746	1.069	3.505	0.45
F0025	2	3	1.919	1.585	1.361	1.124	1.257	1.038	1.64	1.354	1.211	-1.61
F0025	3	0	7.879	2.371	6.708	2.019	3.608	1.086	3.101	0.933	3.323	0.76
F0025	3	1	8.753	2.44	7.304	2.036	3.718	1.036	3.48	0.97	3.587	0.54
F0025	3	2	1.925	1.57	1.384	1.129	1.274	1.039	1.668	1.36	1.226	-1.68
F0026	0	1	2.982	1.483	2.502	1.244	2.026	1.007	2.526	1.256	2.011	-0.58
F0026	0	2	4.356	1.227	4.226	1.19	3.555	1.001	4.133	1.164	3.55	-0.19
F0026	0	3	3.906	1.424	3.831	1.397	2.743	1	3.454	1.259	2.743	-0.04
F0026	1	0	2.991	1.442	2.557	1.233	2.087	1.007	2.296	1.107	2.074	-0.55
F0026	1	2	4.539	1.297	3.85	1.1	3.5	1	3.807	1.088	3.499	0.1
F0026	1	3	3.641	1.236	3.793	1.287	2.974	1.009	3.236	1.098	2.946	0.3
F0026	2	0	4.298	1.194	4.304	1.196	3.608	1.002	4.177	1.16	3.6	-0.21
F0026	2	1	4.427	1.297	3.695	1.083	3.413	1	3.791	1.111	3.412	0.06
F0026	2	3	2.452	1.33	1.969	1.068	1.911	1.036	2.184	1.184	1.844	-3.29
F0026	3	0	3.852	1.401	3.93	1.429	2.75	1	3.442	1.252	2.75	-0.01
F0026	3	1	3.552	1.227	3.694	1.276	2.924	1.01	3.18	1.098	2.895	0.31
F0026	3	2	2.461	1.316	1.975	1.055	1.923	1.028	2.165	1.157	1.871	-3.54
F0027	0	1	1.961	1.052	2.113	1.133	1.878	1.007	1.779	0.954	1.865	0.46
F0027	0	2	3.986	1.108	4.561	1.268	3.67	1.02	3.793	1.055	3.597	0.46
F0027	0	3	4.094	1.073	4.941	1.295	3.975	1.042	3.944	1.034	3.816	0.55
F0027	1	0	1.967	1.057	2.108	1.133	1.873	1.007	1.784	0.958	1.861	0.45
F0027	1	2	3.375	1.112	3.781	1.246	3.064	1.009	3.878	1.278	3.035	0.36
F0027	1	3	3.625	1.115	4.078	1.254	3.305	1.016	4.102	1.262	3.251	0.45
F0027	2	0	3.947	1.091	4.521	1.249	3.675	1.016	4.063	1.123	3.619	0.43
F0027	2	1	3.345	1.086	3.82	1.24	3.105	1.008	4.032	1.309	3.081	0.34
F0027	2	3	1.545	1.293	1.26	1.054	1.198	1.003	1.562	1.307	1.195	-1.18
F0027	3	0	4.064	1.063	4.801	1.256	3.924	1.026	4.231	1.106	3.824	0.52
F0027	3	1	3.606	1.097	4.026	1.225	3.319	1.01	4.256	1.295	3.286	0.37
F0027	3	2	1.545	1.299	1.266	1.065	1.19	1.001	1.508	1.269	1.189	-0.65
F0029	0	1	1.208	1.253	1.045	1.084	0.985	1.022	1.278	1.325	0.964	-1.07
F0029	0	2	3.258	1.082	3.545	1.178	3.01	1	2.879	0.957	3.01	0.08
F0029	0	3	5.151	1.626	3.66	1.155	3.171	1.001	3.375	1.065	3.169	-0.08
F0029	1	0	1.207	1.256	1.046	1.088	0.986	1.026	1.241	1.291	0.961	-1.09
F0029	1	2	3.278	1.112	3.494	1.185	2.982	1.011	2.806	0.952	2.948	0.26
F0029	1	3	5.377	1.637	3.755	1.143	3.405	1.037	3.525	1.073	3.284	0.3
F0029	2	0	3.214	1.058	3.61	1.188	3.045	1.002	2.917	0.96	3.038	0.16
F0029	2	1	3.253	1.106	3.509	1.193	2.983	1.014	2.783	0.946	2.941	0.28
F0029	2	3	4.458	2.37	2.059	1.095	1.907	1.014	2.499	1.328	1.881	-1.05
F0029	3	0	5.141	1.665	3.519	1.14	3.092	1.001	3.165	1.025	3.088	-0.22
F0029	3	1	5.395	1.747	3.58	1.159	3.26	1.055	3.194	1.034	3.088	0.31
F0029	3	2	4.472	2.493	1.985	1.106	1.815	1.012	2.369	1.32	1.794	-0.87

APPENDIX II

FILE FORMAT

II.1 SZE

The Inspeck's proprietary *sze* file format:

```
# The first part is the data of the texture image, each elements in the image
# table corresponds to a 32 bits color value (RGBA) with the hexadecimal format.
# the origin point is at the top left corner, and the order is from left to
# right, and from top to bottom.

BEGIN SZE ASCII
NbNet=1
BEGIN IMAGES
NbImage=1
BEGIN IMAGE "Name of model"
1024 1024
0x505452FF 0x515551FF 0x525753FF
... ... ... ...  (0xRGBA)
0x757E79FF 0x767F7AFF 0x74807AFF 0x74807AFF
END IMAGE "Name of model"
END IMAGES

# And the second part is the polygons' Data that includes:


# The table of polygons' connectivity, each line $i$ contains the point indexes
# (corresponds to the table of the 3D point's coordinate) of the vertexes  of
# the polygon $i$.

BEGIN NET "Name of model"
BEGIN MESH
NbPolygon=88566         #number of polygons
SizePolygon=354264      #4*NbPolygon
3                 #vertices' number
1 167 166         #vertices' index
3
2 165 164
...
...
3
```

```
44461 44470 44479
3
44461 44479 44476
END MESH

# The table of the texture's connectivity, each line $i$ contains the
# indexes of the texture polygons $i$. (corresponds to the table of the 2D
# texture coordinate)

BEGIN MESH_TEX
3
4 167 166
3
5 165 164
...
...
3
44461 44470 44479
END MESH_TEX

# The table of the 2D texture coordinate, the texture's origin is the top
# left corner of the texture image, and the coordinates are normalized. For
# example, in the $1024 \times 1024$ texture image, the coordinate $[0.5,0.5]$
# represents the pixel $x=511,y=511$.

NbTexture=44646   #Number of the texture points
0.255473 0.874281
0.653523 0.875172
...
...
0.919263 0.036554
END TEXTURE

# The table of the 3D points on the mesh.

2 XPos YPos Zpos Row=1 Col=44480 Zmin= 0  File=0
-47.103683 -74.470955 53.607872
166.209793 -75.063454 -33.661514
...
...
71.221840 488.556030 7.318720
END NET "Name of model"
END SZE ASCII
```

The texture is not continuous, it is collection of partial view of the object, make an atlas. it is difficult to think about the relationship between lots of little pieces. the advantage of this atlas texture than the unfold texture image is less distortion on each little piece of atlas, and less wasted space.

APPENDIX III

RELEVANT SOFTWARE

In this project, We have used the many softwares. Most of them are open source software (except FastRBF). Thanks the authors who make these powerful tools available for the public without fee.

III.1 VTK

Web Site: http://www.vtk.org

The Visualization Toolkit (VTK) is an open source, freely available software system for 3D computer graphics, image processing, and visualization used by thousands of researchers and developers around the world. VTK consists of a C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python. Professional support and products for VTK are provided by Kitware, Inc. VTK supports a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques such as implicit modeling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. In addition, dozens of imaging algorithms have been directly integrated to allow the user to mix 2D imaging / 3D graphics algorithms and data. The design and implementation of the library has been strongly influenced by object-oriented principles. VTK has been installed and tested on nearly every Unix-based platform, PCs (Windows 98/ME/NT/2000/XP), and Mac OSX Jaguar or later.

III.2 ParaView

Web Site: http://www.paraview.org

ParaView is an application designed with the need to visualize large data sets in mind.

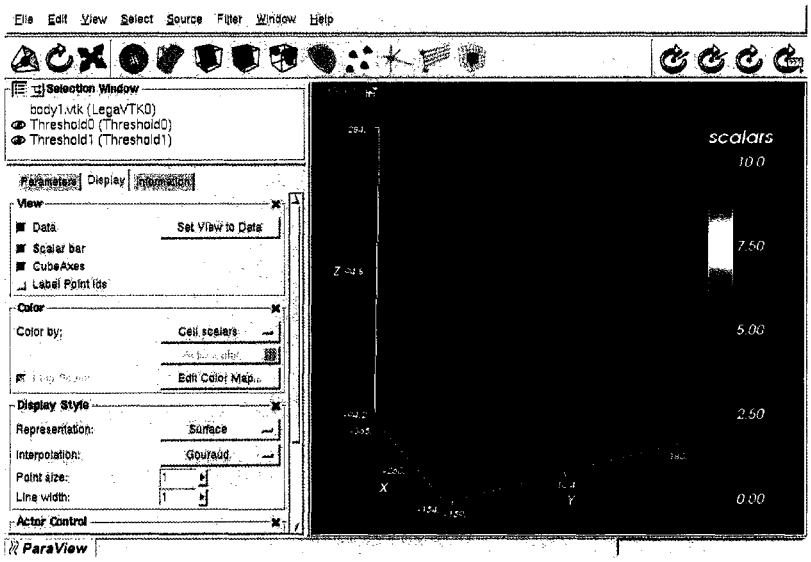


Figure III.1 Screenshot of ParaView — scientific visualization tool.

It is also an open-source, multi-platform visualization application based on the VTK. ParaView runs on distributed and shared memory parallel as well as single processor systems and has been successfully tested on Windows, Linux and various Unix workstations and clusters. Under the hood, ParaView uses the Visualization Toolkit as the data processing and rendering engine and has a user interface written using a unique blend of Tcl/Tk and C++.

We use the paraview the visualize the model in the work. Figure III.1 is a screenshot of Paraview.

III.3 FastRBF

Web Site: http://www.farfieldtechnology.com/products/toolbox/

FastRBF allows RBFs to be fitted to systems with more than 1,000,000 data points on ordinary computing hardware. The polyharmonic (thin-plate) splines that FastRBF uses result in the "smoothest" interpolant, in the sense that they minimize the energy in the 2nd, 3rd, or higher derivatives. This results in extraordinary extrapolation capabilities

when large gaps occur in a data set. In the case of noisy data, FastRBF's error-bar fitter approximates the data with the smoothest function within the error bounds specified at each data point. Low-pass filtering of scattered data can also be achieved directly with FastRBF without resampling on a regular grid and performing computationally intensive convolutions in the spatial or Fourier domains.

We use the evaluation version of FastRBF1.4.1 to fill hole, simplify and smooth mesh.

III.4 NetGen

Web Site: http://www.hpfem.jku.at/netgen/

NETGEN is an automatic 3d tetrahedral mesh generator. It accepts input from constructive solid geometry (CSG) or boundary representation (B-rep) from STL file format. The connection to a geometry kernel allows the handling of IGES and STEP files. NETGEN contains modules for mesh optimization and hierarchical mesh refinement. NetGen is open source based on the LGPL license. It is available for Unix/Linux and Windows.

NETGEN was developed mainly by Joachim Schörl within project grants from the Austrian Science Fund FWF (Special Research Project "Numerical and Symbolic Scientific Computing", Start Project "hp-FEM) at the Johannes Kepler University Linz. Significant contributions were made by Johannes Gerstmayr (STL geometry) Robert Gaisbauer (OpenCascade interface).

The latest version—NetGen4.3.1—was used. And many codes have been written to import and export the different file formats to work with NetGen.

III.5 GTS

Web Site: http://gts.sourceforge.net/

GTS stands for the GNU Triangulated Surface Library. It is an Open Source Free Software Library intended to provide a set of useful functions to deal with 3D surfaces meshed with interconnected triangles. The source code is available free of charge un-

der the Free Software LGPL license. The code is written entirely in C with an object-oriented approach based mostly on the design of GTK+. Careful attention is paid to performance related issues as the initial goal of GTS is to provide a simple and efficient library to scientists dealing with 3D computational surface meshes.

III.6 GSL

Web Site: http://www.gnu.org/software/gsl/

The GNU Scientific Library (GSL) is a numerical library for C and C++ programmers. The current version is GSL-1.4. It was released on 14 August 2003. GSL provides a well-defined C language Applications Programming Interface (API) for common numerical functions, while allowing wrappers to be written for very high level languages.The routines have been written from scratch by the GSL team in ANSI C.

III.7 Wings3D

Web Site: http://www.wings3d.com/

An open source interactive 3D modeling tools. We use it to manipulate the bone model manually to remove the undesired points, edges, and triangles to fix the surface mesh.

III.8 SQLite

Web Site: http://www.sqlite.org/

SQLite is a C/C++ library that implements an embeddable SQL database engine. Programs that link with the SQLite library can have SQL database access without running a separate RDBMS process. The distribution comes with a standalone command-line access program (sqlite) that can be used to administer an SQLite database and which serves as an example of how to use the SQLite library. SQLite is not a client library used to connect to a big database server. SQLite is the server. The SQLite library reads

and writes directly to and from the database files on disk. Its sources are in the public domain. And we can use it for any purpose.

We use SQLite to store and analyze our experimental data.

III.9 QT

Web Site: http://www.trolltech.com/

QT is a powerful user interface package for C++. It is open source and free available.

We used the QT to write the GUI for the program of picking landmarks' position on the human external skin surface mesh model from Inspeck's 3D Camera.

III.10 Linux & gcc

All the codes in the thesis work were written on the Linux platform with C++ (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5)).

III.11 Kile

Web Site: http://kile.sourceforge.net/

The aim of the Kile project is to develop a user friendly TeX/LaTeX editor. It will run on systems with the KDE desktop environment. KDE is available for many architectures with Linux installed.